Randomised Algorithms

Lecture 2-3: Concentration Inequalities

Thomas Sauerwald (tms41@cam.ac.uk)

Lent 2022

UNIVERSITY OF CAMBRIDGE

Outline

Concentration Inequalities

- Concentration refers to the phenomena where random variables are very close to their mean

Concentration Inequalities

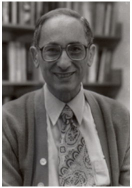
- Concentration refers to the phenomena where random variables are very close to their mean
- This is very useful in randomised algorithms as it ensures an **almost** deterministic behaviour

Concentration Inequalities

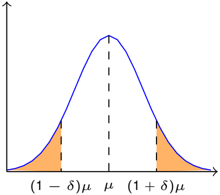
- Concentration refers to the phenomena where random variables are very close to their mean
- This is very useful in randomised algorithms as it ensures an **almost** deterministic behaviour
- It gives us the best of two worlds:
  1. **Randomised Algorithms:** Easy to Design and Implement
  2. **Deterministic Algorithms:** They do what they claim

Chernoff Bounds: A Tool for Concentration

- Chernoffs bounds are "strong" bounds on the tail probabilities of sums of independent random variables
- random variables can be discrete (or continuous)
- usually these bounds decrease exponentially as opposed to a polynomial decrease in Markov's or Chebyshev's inequality (see example)

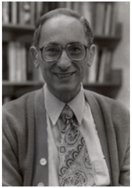


Hermann Chernoff (1923-)

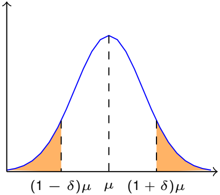


Chernoff Bounds: A Tool for Concentration

- Chernoffs bounds are "strong" bounds on the tail probabilities of sums of independent random variables
- random variables can be discrete (or continuous)
- usually these bounds decrease exponentially as opposed to a polynomial decrease in Markov's or Chebyshev's inequality (see example)
- easy to apply, but requires independence



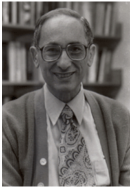
Hermann Chernoff (1923-)



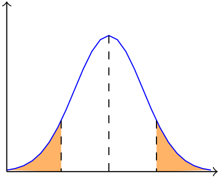
Chernoff Bounds: A Tool for Concentration

- Chernoffs bounds are “strong” bounds on the tail probabilities of sums of independent random variables
- random variables can be discrete (or continuous)
- usually these bounds decrease exponentially as opposed to a polynomial decrease in Markov’s or Chebyshev’s inequality (see example)
- easy to apply, but requires independence
- have found various applications in:
  - Randomised Algorithms
  - Statistics
  - Random Projections and Dimensionality Reduction
  - Learning Theory (e.g., PAC-learning)

  ⋮



Hermann Chernoff (1923-)



Recap: Markov and Chebyshev

Markov's Inequality

If X is a non-negative random variable, then for any $a > 0$,

$$\mathbf{P}[X \geq a] \leq \mathbf{E}[X]/a.$$

Chebyshev's Inequality

If X is a random variable, then for any $a > 0$,

$$\mathbf{P}[|X - \mathbf{E}[X]| \geq a] \leq \mathbf{V}[X]/a^2.$$

Recap: Markov and Chebyshev

Markov's Inequality

If X is a non-negative random variable, then for any $a > 0$,

$$\mathbf{P}[X \geq a] \leq \mathbf{E}[X]/a.$$

Chebyshev's Inequality

If X is a random variable, then for any $a > 0$,

$$\mathbf{P}[|X - \mathbf{E}[X]| \geq a] \leq \mathbf{V}[X]/a^2.$$

- Let $f : \mathbb{R} \to [0, \infty)$ and increasing, then $f(X) \geq 0$, and thus

$$\mathbf{P}[X \geq a] \leq \mathbf{P}[f(X) \geq f(a)] \leq \mathbf{E}[f(X)]/f(a).$$

Recap: Markov and Chebyshev

Markov's Inequality

If X is a non-negative random variable, then for any $a > 0$,

$$\mathbf{P}[X \geq a] \leq \mathbf{E}[X]/a.$$

Chebyshev's Inequality

If X is a random variable, then for any $a > 0$,

$$\mathbf{P}[|X - \mathbf{E}[X]| \geq a] \leq \mathbf{V}[X]/a^2.$$

- Let $f : \mathbb{R} \to [0, \infty)$ and increasing, then $f(X) \geq 0$, and thus

$$\mathbf{P}[X \geq a] \leq \mathbf{P}[f(X) \geq f(a)] \leq \mathbf{E}[f(X)]/f(a).$$

- Similarly, if $g : \mathbb{R} \to [0, \infty)$ and decreasing, then $g(X) \geq 0$, and thus

$$\mathbf{P}[X \leq a] \leq \mathbf{P}[g(X) \geq g(a)] \leq \mathbf{E}[g(X)]/g(a).$$

Recap: Markov and Chebyshev

Markov's Inequality

If X is a non-negative random variable, then for any $a > 0$,

$$\mathbf{P}[X \geq a] \leq \mathbf{E}[X]/a.$$

Chebyshev's Inequality

If X is a random variable, then for any $a > 0$,

$$\mathbf{P}[|X - \mathbf{E}[X]| \geq a] \leq \mathbf{V}[X]/a^2.$$

- Let $f : \mathbb{R} \to [0, \infty)$ and increasing, then $f(X) \geq 0$, and thus

$$\mathbf{P}[X \geq a] \leq \mathbf{P}[f(X) \geq f(a)] \leq \mathbf{E}[f(X)]/f(a).$$

- Similarly, if $g : \mathbb{R} \to [0, \infty)$ and decreasing, then $g(X) \geq 0$, and thus

$$\mathbf{P}[X \leq a] \leq \mathbf{P}[g(X) \geq g(a)] \leq \mathbf{E}[g(X)]/g(a).$$

Chebyshev's inequality (or Markov) can be obtained by chosing $f(X) := (X - \mu)^2$ (or $f(X) := X$, respectively).

From Markov and Chebyshev to Chernoff

Markov and Chebyshev use the first and second moment of the random variable. Can we keep going?

From Markov and Chebyshev to Chernoff

Markov and Chebyshev use the first and second moment of the random variable. Can we keep going?

- Yes!

From Markov and Chebyshev to Chernoff

Markov and Chebyshev use the first and second moment of the random variable. Can we keep going?

- Yes!

We can consider the first, second, third and more moments! That is the basic idea behind the Chernoff Bounds

Our First Chernoff Bound

Chernoff Bounds (General Form, Upper Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}. \qquad (\star)$$

Our First Chernoff Bound

Chernoff Bounds (General Form, Upper Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}. \qquad (\star)$$

While $(\star)$ is one of the easiest (and most generic) Chernoff bounds to derive, the bound is complicated and hard to apply...

Our First Chernoff Bound

Chernoff Bounds (General Form, Upper Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}. \qquad (\bigstar)$$

This implies that for any $t > \mu$,

$$\mathbf{P}[X \geq t] \leq e^{-\mu}\left(\frac{e\mu}{t}\right)^{t}.$$

While ($\bigstar$) is one of the easiest (and most generic) Chernoff bounds to derive, the bound is complicated and hard to apply...

Example: Coin Flips (1/3)

- Consider throwing a fair coin n times and count the total number of heads

Example: Coin Flips (1/3)

- Consider throwing a fair coin n times and count the total number of heads
- $X_i \in \{0, 1\}$, $X = \sum_{i=1}^{n} X_i$ and $\mathbf{E}[X] = n \cdot 1/2 = n/2$

Example: Coin Flips (1/3)

- Consider throwing a fair coin n times and count the total number of heads
- $X_i \in \{0, 1\}$, $X = \sum_{i=1}^{n} X_i$ and $\mathbf{E}[X] = n \cdot 1/2 = n/2$
- The Chernoff Bound gives for any $\delta > 0$,

$$\mathbf{P}[X \geq (1+\delta)(n/2)] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{n/2}.$$

Example: Coin Flips (1/3)

- Consider throwing a fair coin n times and count the total number of heads
- $X_i \in \{0, 1\}$, $X = \sum_{i=1}^{n} X_i$ and $\mathbf{E}[X] = n \cdot 1/2 = n/2$
- The Chernoff Bound gives for any $\delta > 0$,

$$\mathbf{P}[X \geq (1+\delta)(n/2)] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{n/2}.$$

- The above expression equals 1 only for $\delta = 0$, and then it gives a value strictly less than 1 (**check this!**)

Example: Coin Flips (1/3)

- Consider throwing a fair coin n times and count the total number of heads
- $X_i \in \{0, 1\}$, $X = \sum_{i=1}^{n} X_i$ and $\mathbf{E}[X] = n \cdot 1/2 = n/2$
- The Chernoff Bound gives for any $\delta > 0$,

$$\mathbf{P}[X \geq (1+\delta)(n/2)] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{n/2}.$$

- The above expression equals 1 only for $\delta = 0$, and then it gives a value strictly less than 1 (**check this!**)
- The inequality is **exponential in** n, (for fixed δ) which is much better than Chebyshev's inequality.

Example: Coin Flips (1/3)

- Consider throwing a fair coin n times and count the total number of heads
- $X_i \in \{0, 1\}$, $X = \sum_{i=1}^{n} X_i$ and $\mathbf{E}[X] = n \cdot 1/2 = n/2$
- The Chernoff Bound gives for any $\delta > 0$,

$$\mathbf{P}[X \geq (1+\delta)(n/2)] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}} \right]^{n/2}.$$

- The above expression equals 1 only for $\delta = 0$, and then it gives a value strictly less than 1 (**check this!**)
- The inequality is **exponential in** n, (for fixed δ) which is much better than Chebyshev's inequality.

What about a concrete value of n, say $n = 100$?

Example: Coin Flips (2/3)

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

Example: Coin Flips (2/3)

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $\mathbf{E}[X] = 100/2 = 50$.

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq 2/3 = \mathbf{0.666}.$$

Example: Coin Flips (2/3)

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $\mathbf{E}[X] = 100/2 = 50$.

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq 2/3 = \mathbf{0.666}.$$

- Chebyshev's inequality: $\mathbf{V}[X] = \sum_{i=1}^{100} \mathbf{V}[X_i] = 100 \cdot (1/2)^2 = 25$.

$$\mathbf{P}[|X - \mu| \geq t] \leq \frac{\mathbf{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = \mathbf{0.04}$, much better than what we obtained by Markov's inequality.

Example: Coin Flips (2/3)

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $\mathbf{E}[X] = 100/2 = 50$.

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq 2/3 = \mathbf{0.666}.$$

- Chebyshev's inequality: $\mathbf{V}[X] = \sum_{i=1}^{100} \mathbf{V}[X_i] = 100 \cdot (1/2)^2 = 25$.

$$\mathbf{P}[|X - \mu| \geq t] \leq \frac{\mathbf{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = \mathbf{0.04}$, much better than what we obtained by Markov's inequality.

- The Chernoff bound: with $\delta = 1/2$ gives:

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq \left(\frac{e^{1/2}}{(3/2)^{3/2}} \right)^{50} = \mathbf{0.004472}.$$

Example: Coin Flips (2/3)

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $\mathbf{E}[X] = 100/2 = 50$.

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq 2/3 = \mathbf{0.666}.$$

- Chebyshev's inequality: $\mathbf{V}[X] = \sum_{i=1}^{100} \mathbf{V}[X_i] = 100 \cdot (1/2)^2 = 25$.

$$\mathbf{P}[|X - \mu| \geq t] \leq \frac{\mathbf{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = \mathbf{0.04}$, much better than what we obtained by Markov's inequality.

- The Chernoff bound: with $\delta = 1/2$ gives:

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq \left(\frac{e^{1/2}}{(3/2)^{3/2}} \right)^{50} = \mathbf{0.004472}.$$

- Remark: The exact probability is **0.00000028 ...**

Example: Coin Flips (2/3)

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $\mathbf{E}[X] = 100/2 = 50$.

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq 2/3 = \mathbf{0.666}.$$

- Chebyshev's inequality: $\mathbf{V}[X] = \sum_{i=1}^{100} \mathbf{V}[X_i] = 100 \cdot (1/2)^2 = 25$.

$$\mathbf{P}[|X - \mu| \geq t] \leq \frac{\mathbf{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = \mathbf{0.04}$, much better than what we obtained by Markov's inequality.

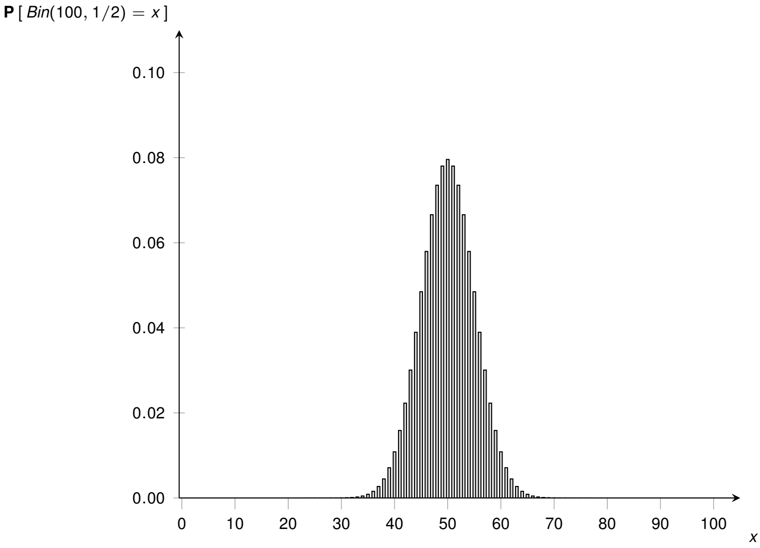
- The Chernoff bound: with $\delta = 1/2$ gives:

$$\mathbf{P}[X \geq 3/2 \cdot \mathbf{E}[X]] \leq \left(\frac{e^{1/2}}{(3/2)^{3/2}} \right)^{50} = \mathbf{0.004472}.$$

- Remark: The exact probability is **0.00000028 . . .**

Chernoff bound yields a much better result (but needs independence!)

Example: Coin Flips (3/3)



Outline

General Recipe for Deriving Chernoff Bounds

Recipe

The three main steps in deriving Chernoff bounds for sums of independent random variables $X = X_1 + \cdots + X_n$ are:

General Recipe for Deriving Chernoff Bounds

Recipe

The three main steps in deriving Chernoff bounds for sums of independent random variables $X = X_1 + \cdots + X_n$ are:

1. Instead of working with X, we switch to the **moment generating function** $e^{\lambda X}$, $\lambda > 0$ and apply Markov's inequality $\rightsquigarrow$ $\mathbf{E}\left[e^{\lambda X} \right]$

General Recipe for Deriving Chernoff Bounds

Recipe

The three main steps in deriving Chernoff bounds for sums of independent random variables $X = X_1 + \cdots + X_n$ are:

1. Instead of working with X, we switch to the **moment generating function** $e^{\lambda X}$, $\lambda > 0$ and apply Markov's inequality $\rightsquigarrow$ $\mathbf{E}\left[e^{\lambda X} \right]$
2. Compute an upper bound for $\mathbf{E}\left[e^{\lambda X} \right]$ (using independence)

General Recipe for Deriving Chernoff Bounds

Recipe

The three main steps in deriving Chernoff bounds for sums of independent random variables $X = X_1 + \cdots + X_n$ are:

1. Instead of working with X, we switch to the **moment generating function** $e^{\lambda X}$, $\lambda > 0$ and apply Markov's inequality $\leadsto$ $\mathbf{E}\left[e^{\lambda X} \right]$
2. Compute an upper bound for $\mathbf{E}\left[e^{\lambda X} \right]$ (using independence)
3. Optimise value of λ to obtain best tail bound

Chernoff Bound: Proof

Chernoff Bound (General Form, Upper Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}.$$

Proof:

Chernoff Bound: Proof

Chernoff Bound (General Form, Upper Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}.$$

Proof:

1. For $\lambda > 0$,

$$\mathbf{P}[X \geq (1+\delta)\mu] \underset{e^{\lambda x} \text{ is incr}}{\leq} \mathbf{P}\left[e^{\lambda X} \geq e^{\lambda(1+\delta)\mu}\right] \underset{\text{Markov}}{\leq} e^{-\lambda(1+\delta)\mu}\mathbf{E}\left[e^{\lambda X}\right]$$

Chernoff Bound: Proof

Chernoff Bound (General Form, Upper Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}.$$

Proof:

1. For $\lambda > 0$,

$$\mathbf{P}[X \geq (1+\delta)\mu] \underset{e^{\lambda x} \text{ is incr}}{\leq} \mathbf{P}\left[e^{\lambda X} \geq e^{\lambda(1+\delta)\mu}\right] \underset{\text{Markov}}{\leq} e^{-\lambda(1+\delta)\mu}\mathbf{E}\left[e^{\lambda X}\right]$$

2. $\mathbf{E}\left[e^{\lambda X}\right] = \mathbf{E}\left[e^{\lambda \sum_{i=1}^n X_i}\right] \underset{\text{indep}}{=} \prod_{i=1}^n \mathbf{E}\left[e^{\lambda X_i}\right]$

Chernoff Bound: Proof

Chernoff Bound (General Form, Upper Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^\mu .$$

Proof:

1. For $\lambda > 0$,

$$\mathbf{P}[X \geq (1+\delta)\mu] \underset{e^{\lambda x} \text{ is incr}}{\leq} \mathbf{P}\left[e^{\lambda X} \geq e^{\lambda(1+\delta)\mu}\right] \underset{\text{Markov}}{\leq} e^{-\lambda(1+\delta)\mu}\mathbf{E}\left[e^{\lambda X}\right]$$

2. $\mathbf{E}\left[e^{\lambda X}\right] = \mathbf{E}\left[e^{\lambda \sum_{i=1}^n X_i}\right] \underset{\text{indep}}{=} \prod_{i=1}^n \mathbf{E}\left[e^{\lambda X_i}\right]$

3. $$\mathbf{E}\left[e^{\lambda X_i}\right] = e^\lambda p_i + (1-p_i) = 1 + p_i(e^\lambda - 1) \underset{1+x \leq e^x}{\leq} e^{p_i(e^\lambda - 1)}$$

Chernoff Bound: Proof

Chernoff Bound: Proof

1. For $\lambda > 0$,

$$\mathbf{P}[X \geq (1+\delta)\mu] \underset{e^{\lambda x} \text{ is incr}}{=} \mathbf{P}\left[e^{\lambda X} \geq e^{\lambda(1+\delta)\mu}\right] \underset{\text{Markov}}{\leq} e^{-\lambda(1+\delta)\mu}\mathbf{E}\left[e^{\lambda X}\right]$$

2. $\mathbf{E}\left[e^{\lambda X}\right] = \mathbf{E}\left[e^{\lambda \sum_{i=1}^{n} X_i}\right] \underset{\text{indep}}{=} \prod_{i=1}^{n} \mathbf{E}\left[e^{\lambda X_i}\right]$

3. $$\mathbf{E}\left[e^{\lambda X_i}\right] = e^{\lambda} p_i + (1-p_i) = 1 + p_i(e^{\lambda}-1) \underset{1+x \leq e^x}{\leq} e^{p_i(e^{\lambda}-1)}$$

Chernoff Bound: Proof

1. For $\lambda > 0$,

$$\mathbf{P}[X \geq (1+\delta)\mu] \underset{e^{\lambda x} \text{ is incr}}{=} \mathbf{P}\left[e^{\lambda X} \geq e^{\lambda(1+\delta)\mu}\right] \underset{\text{Markov}}{\leq} e^{-\lambda(1+\delta)\mu}\mathbf{E}\left[e^{\lambda X}\right]$$

2. $\mathbf{E}\left[e^{\lambda X}\right] = \mathbf{E}\left[e^{\lambda \sum_{i=1}^{n} X_i}\right] \underset{\text{indep}}{=} \prod_{i=1}^{n} \mathbf{E}\left[e^{\lambda X_i}\right]$

3. $$\mathbf{E}\left[e^{\lambda X_i}\right] = e^{\lambda} p_i + (1-p_i) = 1 + p_i(e^{\lambda} - 1) \underset{1+x \leq e^x}{\leq} e^{p_i(e^{\lambda}-1)}$$

4. Putting all together

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq e^{-\lambda(1+\delta)\mu} \prod_{i=1}^{n} e^{p_i(e^{\lambda}-1)} = e^{-\lambda(1+\delta)\mu} e^{\mu(e^{\lambda}-1)}$$

Chernoff Bound: Proof

1. For $\lambda > 0$,

$$\mathbf{P}[X \geq (1+\delta)\mu] \underset{e^{\lambda x} \text{ is incr}}{=} \mathbf{P}\left[e^{\lambda X} \geq e^{\lambda(1+\delta)\mu}\right] \underset{\text{Markov}}{\leq} e^{-\lambda(1+\delta)\mu}\mathbf{E}\left[e^{\lambda X}\right]$$

2. $\mathbf{E}\left[e^{\lambda X}\right] = \mathbf{E}\left[e^{\lambda \sum_{i=1}^n X_i}\right] \underset{\text{indep}}{=} \prod_{i=1}^n \mathbf{E}\left[e^{\lambda X_i}\right]$

3. $$\mathbf{E}\left[e^{\lambda X_i}\right] = e^{\lambda}p_i + (1-p_i) = 1 + p_i(e^{\lambda}-1) \underset{1+x \leq e^x}{\leq} e^{p_i(e^{\lambda}-1)}$$

4. Putting all together

$$\mathbf{P}[X \geq (1+\delta)\mu] \leq e^{-\lambda(1+\delta)\mu} \prod_{i=1}^n e^{p_i(e^{\lambda}-1)} = e^{-\lambda(1+\delta)\mu} e^{\mu(e^{\lambda}-1)}$$

5. Choose $\lambda = \log(1+\delta) > 0$ to get the result.

Chernoff Bounds: Lower Tails

We can also use Chernoff Bounds to show a random variable is **not too small** compared to its mean:

Chernoff Bounds (General Form, Lower Tail)

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum p_i$. Then, for any $\delta > 0$ it holds that

$$\mathbf{P}[X \leq (1-\delta)\mu] \leq \left[\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right]^{\mu},$$

and thus, by substitution, for any $t < \mu$,

$$\mathbf{P}[X \leq t] \leq e^{-\mu}\left(\frac{e\mu}{t}\right)^{t}.$$

Exercise on Supervision Sheet

Hint: multiply both sides by -1 and repeat the proof of the Chernoff Bound

Nicer Chernoff Bounds

Nicer Chernoff Bounds

"Nicer" Chernoff Bounds

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. Then,

Nicer Chernoff Bounds

"Nicer" Chernoff Bounds

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$. Then,

- For all $t > 0$,

$$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$$

$$\mathbf{P}[X \leq \mathbf{E}[X] - t] \leq e^{-2t^2/n}$$

Nicer Chernoff Bounds

"Nicer" Chernoff Bounds

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. Then,

- For all $t > 0$,

$$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$$

$$\mathbf{P}[X \leq \mathbf{E}[X] - t] \leq e^{-2t^2/n}$$

- For $0 < \delta < 1$,

$$\mathbf{P}[X \geq (1+\delta)\mathbf{E}[X]] \leq \exp\left(-\frac{\delta^2 \mathbf{E}[X]}{3}\right)$$

$$\mathbf{P}[X \leq (1-\delta)\mathbf{E}[X]] \leq \exp\left(-\frac{\delta^2 \mathbf{E}[X]}{2}\right)$$

Nicer Chernoff Bounds

"Nicer" Chernoff Bounds

Suppose $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameter p_i. Let $X = X_1 + \ldots + X_n$ and $\mu = \mathbf{E}[X] = \sum_{i=1}^n p_i$. Then,

- For all $t > 0$,

$$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$$

$$\mathbf{P}[X \leq \mathbf{E}[X] - t] \leq e^{-2t^2/n}$$

- For $0 < \delta < 1$,

$$\mathbf{P}[X \geq (1+\delta)\mathbf{E}[X]] \leq \exp\left(-\frac{\delta^2 \mathbf{E}[X]}{3}\right)$$

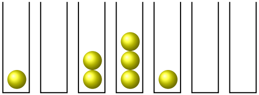
$$\mathbf{P}[X \leq (1-\delta)\mathbf{E}[X]] \leq \exp\left(-\frac{\delta^2 \mathbf{E}[X]}{2}\right)$$

All upper tail bounds hold even under a relaxed independence assumption: For all $1 \leq i \leq n$ and $x_1, x_2, \ldots, x_{i-1} \in \{0, 1\}$,

$$\mathbf{P}[X_i = 1 \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1}] \leq p_i.$$

Outline

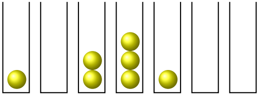
Balls into Bins



Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Balls into Bins

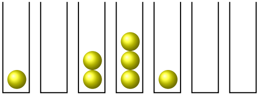


Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

- A very natural but also rich mathematical model

Balls into Bins

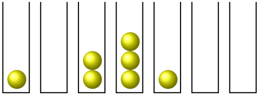


Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

- A very natural but also rich mathematical model
- In computer science, there are several interpretations:

Balls into Bins

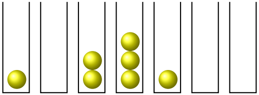


Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

- A very natural but also rich mathematical model
- In computer science, there are several interpretations:
  1. Bins are a hash table, balls are items
  2. Bins are processors and balls are jobs
  3. Bins are data servers and balls are queries

Balls into Bins



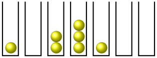
Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

- A very natural but also rich mathematical model
- In computer science, there are several interpretations:
  1. Bins are a hash table, balls are items
  2. Bins are processors and balls are jobs
  3. Bins are data servers and balls are queries

Exercise: Think about the relation between the Balls into Bins Model and the Coupon Collector Problem.

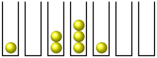
Balls into Bins: Bounding the Maximum Load (1/4)



Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Balls into Bins: Bounding the Maximum Load (1/4)

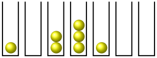


Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Question 1: How large is the maximum load if $m = 2n \log n$?

Balls into Bins: Bounding the Maximum Load (1/4)



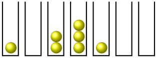
Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Question 1: How large is the maximum load if $m = 2n \log n$?

- Focus on an arbitrary single bin. Let X_i the indicator variable which is 1 iff ball i is assigned to this bin. Note that $p_i = \mathbf{P}[X_i = 1] = 1/n$.

Balls into Bins: Bounding the Maximum Load (1/4)



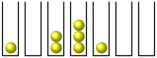
Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Question 1: How large is the maximum load if $m = 2n \log n$?

- Focus on an arbitrary single bin. Let X_i the indicator variable which is 1 iff ball i is assigned to this bin. Note that $p_i = \mathbf{P}[X_i = 1] = 1/n$.
- The total balls in the bin is given by $X := \sum_{i=1}^{n} X_i$.

Balls into Bins: Bounding the Maximum Load (1/4)



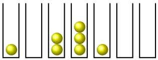
Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Question 1: How large is the maximum load if $m = 2n \log n$?

- Focus on an arbitrary single bin. Let X_i the indicator variable which is 1 iff ball i is assigned to this bin. Note that $p_i = \mathbf{P}[X_i = 1] = 1/n$.
- The total balls in the bin is given by $X := \sum_{i=1}^{n} X_i$.
- Since $m = 2n \log n$, then $\mu = \mathbf{E}[X] = 2 \log n$

Balls into Bins: Bounding the Maximum Load (1/4)



Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Question 1: How large is the maximum load if $m = 2n \log n$?

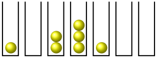
- Focus on an arbitrary single bin. Let X_i the indicator variable which is 1 iff ball i is assigned to this bin. Note that $p_i = \mathbf{P}[X_i = 1] = 1/n$.
- The total balls in the bin is given by $X := \sum_{i=1}^{n} X_i$.
- Since $m = 2n \log n$, then $\mu = \mathbf{E}[X] = 2 \log n$

$$\mathbf{P}[X \geq t] \leq e^{-\mu}(e\mu/t)^t$$

- By the Chernoff Bound,

$$\mathbf{P}[X \geq 6 \log n] \leq e^{-2\log n}\left(\frac{2e \log n}{6 \log n}\right)^{6 \log n} \leq e^{-2\log n} = n^{-2}$$

Balls into Bins: Bounding the Maximum Load (1/4)



Balls into Bins Model

You have m balls and n bins. Each ball is allocated in a bin picked independently and uniformly at random.

Question 1: How large is the maximum load if $m = 2n \log n$?

- Focus on an arbitrary single bin. Let X_i the indicator variable which is 1 iff ball i is assigned to this bin. Note that $p_i = \mathbf{P}[X_i = 1] = 1/n$.
- The total balls in the bin is given by $X := \sum_{i=1}^{n} X_i$.
- Since $m = 2n \log n$, then $\mu = \mathbf{E}[X] = 2 \log n$

here we could have used the "nicer" bounds as well!

$$\mathbf{P}[X \geq t] \leq e^{-\mu}(e\mu/t)^t$$

- By the Chernoff Bound,

$$\mathbf{P}[X \geq 6 \log n] \leq e^{-2 \log n} \left(\frac{2e \log n}{6 \log n}\right)^{6 \log n} \leq e^{-2 \log n} = n^{-2}$$

Balls into Bins: Bounding the Maximum Load (2/4)

- Let $\mathcal{E}_j := \{X(j) \geq 6 \log n\}$, that is, bin j receives at least $6 \log n$ balls.

Balls into Bins: Bounding the Maximum Load (2/4)

- Let $\mathcal{E}_j := \{X(j) \geq 6 \log n\}$, that is, bin j receives at least $6 \log n$ balls.
- We are interested in the probability that at least one bin receives at least $6 \log n$ balls $\Rightarrow$ this is the event $\bigcup_{j=1}^{n} \mathcal{E}_j$

Balls into Bins: Bounding the Maximum Load (2/4)

- Let $\mathcal{E}_j := \{X(j) \geq 6 \log n\}$, that is, bin j receives at least $6 \log n$ balls.
- We are interested in the probability that at least one bin receives at least $6 \log n$ balls $\Rightarrow$ this is the event $\bigcup_{j=1}^{n} \mathcal{E}_j$
- By the Union Bound,

$$\mathbf{P}\left[\bigcup_{j=1}^{n} \mathcal{E}_j\right] \leq \sum_{j=1}^{n} \mathbf{P}\left[\,\mathcal{E}_j\,\right] \leq n \cdot n^{-2} = n^{-1}.$$

Balls into Bins: Bounding the Maximum Load (2/4)

- Let $\mathcal{E}_j := \{X(j) \geq 6 \log n\}$, that is, bin j receives at least $6 \log n$ balls.
- We are interested in the probability that at least one bin receives at least $6 \log n$ balls $\Rightarrow$ this is the event $\bigcup_{j=1}^{n} \mathcal{E}_j$
- By the Union Bound,

$$\mathbf{P}\left[\bigcup_{j=1}^{n} \mathcal{E}_j\right] \leq \sum_{j=1}^{n} \mathbf{P}\left[\,\mathcal{E}_j\,\right] \leq n \cdot n^{-2} = n^{-1}.$$

- Therefore whp, no bin receives at least $6 \log n$ balls

Balls into Bins: Bounding the Maximum Load (2/4)

- Let $\mathcal{E}_j := \{X(j) \geq 6 \log n\}$, that is, bin j receives at least $6 \log n$ balls.
- We are interested in the probability that at least one bin receives at least $6 \log n$ balls $\Rightarrow$ this is the event $\bigcup_{j=1}^{n} \mathcal{E}_j$
- By the Union Bound,

$$\mathbf{P}\left[\bigcup_{j=1}^{n} \mathcal{E}_j\right] \leq \sum_{j=1}^{n} \mathbf{P}\left[\mathcal{E}_j\right] \leq n \cdot n^{-2} = n^{-1}.$$

- Therefore whp, no bin receives at least $6 \log n$ balls

whp stands for *with high probability*:
An event $\mathcal{E}$ (that implicitly depends on an input parameter n) occurs whp if
$\mathbf{P}[\mathcal{E}] \to 1$ as $n \to \infty$.
This is a very standard notation in randomised algorithms
but it may vary from author to author. Be careful!

Balls into Bins: Bounding the Maximum Load (2/4)

- Let $\mathcal{E}_j := \{X(j) \geq 6 \log n\}$, that is, bin j receives at least $6 \log n$ balls.
- We are interested in the probability that at least one bin receives at least $6 \log n$ balls $\Rightarrow$ this is the event $\bigcup_{j=1}^{n} \mathcal{E}_j$
- By the Union Bound,

$$\mathbf{P}\left[\bigcup_{j=1}^{n} \mathcal{E}_j\right] \leq \sum_{j=1}^{n} \mathbf{P}\left[\mathcal{E}_j\right] \leq n \cdot n^{-2} = n^{-1}.$$

- Therefore whp, no bin receives at least $6 \log n$ balls
- By pigeonhole principle, the max loaded bin receives at least $2 \log n$ balls. Hence our bound is pretty sharp.

whp stands for *with high probability*:
An event $\mathcal{E}$ (that implicitly depends on an input parameter n) occurs whp if $\mathbf{P}\left[\mathcal{E}\right] \to 1$ as $n \to \infty$.
This is a very standard notation in randomised algorithms but it may vary from author to author. Be careful!

Balls into Bins: Bounding the Maximum Load (3/4)

Question 2: How large is the maximum load if $m = n$?

Balls into Bins: Bounding the Maximum Load (3/4)

Question 2: How large is the maximum load if $m = n$?

- Using the Chernoff Bound:

$$\mathbf{P}[X \geq t] \leq e^{-\mu}(e\mu/t)^t$$

$$\mathbf{P}[X \geq t] \leq e^{-1}\left(\frac{e}{t}\right)^t \leq \left(\frac{e}{t}\right)^t$$

Balls into Bins: Bounding the Maximum Load (3/4)

Question 2: How large is the maximum load if $m = n$?

- Using the Chernoff Bound:

$$\mathbf{P}[X \geq t] \leq e^{-\mu}(e\mu/t)^t$$

$$\mathbf{P}[X \geq t] \leq e^{-1}\left(\frac{e}{t}\right)^t \leq \left(\frac{e}{t}\right)^t$$

- By setting $t = 4 \log n / \log\log n$, we claim to obtain $\mathbf{P}[X \geq t] \leq n^{-2}$.

Question 2: How large is the maximum load if $m = n$?

- Using the Chernoff Bound:

$$\mathbf{P}[X \geq t] \leq e^{-\mu}(e\mu/t)^t$$

$$\mathbf{P}[X \geq t] \leq e^{-1}\left(\frac{e}{t}\right)^t \leq \left(\frac{e}{t}\right)^t$$

- By setting $t = 4\log n / \log\log n$, we claim to obtain $\mathbf{P}[X \geq t] \leq n^{-2}$.
- Indeed:

$$\left(\frac{e\log\log n}{4\log n}\right)^{4\log n/\log\log n} = \exp\left(\frac{4\log n}{\log\log n}\cdot\log\left(\frac{e\log\log n}{4\log n}\right)\right)$$

Balls into Bins: Bounding the Maximum Load (3/4)

Question 2: How large is the maximum load if $m = n$?

- Using the Chernoff Bound: $\mathbf{P}[X \geq t] \leq e^{-\mu}(e\mu/t)^t$

$$\mathbf{P}[X \geq t] \leq e^{-1}\left(\frac{e}{t}\right)^t \leq \left(\frac{e}{t}\right)^t$$

- By setting $t = 4\log n / \log\log n$, we claim to obtain $\mathbf{P}[X \geq t] \leq n^{-2}$.
- Indeed:

$$\left(\frac{e\log\log n}{4\log n}\right)^{4\log n/\log\log n} = \exp\left(\frac{4\log n}{\log\log n}\cdot\log\left(\frac{e\log\log n}{4\log n}\right)\right)$$

- The term inside the exponential is

$$\frac{4\log n}{\log\log n}\cdot(\log(4/e) + \log\log\log n - \log\log n)$$

Balls into Bins: Bounding the Maximum Load (3/4)

Question 2: How large is the maximum load if $m = n$?

- Using the Chernoff Bound:

$$\mathbf{P}[X \geq t] \leq e^{-\mu}(e\mu/t)^t$$

$$\mathbf{P}[X \geq t] \leq e^{-1}\left(\frac{e}{t}\right)^t \leq \left(\frac{e}{t}\right)^t$$

- By setting $t = 4\log n/\log\log n$, we claim to obtain $\mathbf{P}[X \geq t] \leq n^{-2}$.
- Indeed:

$$\left(\frac{e\log\log n}{4\log n}\right)^{4\log n/\log\log n} = \exp\left(\frac{4\log n}{\log\log n}\cdot\log\left(\frac{e\log\log n}{4\log n}\right)\right)$$

- The term inside the exponential is

$$\frac{4\log n}{\log\log n}\cdot(\log(4/e)+\log\log\log n-\log\log n) \leq \frac{4\log n}{\log\log n}\left(-\frac{1}{2}\log\log n\right),$$

This inequality only works for large enough n.

Balls into Bins: Bounding the Maximum Load (3/4)

Question 2: How large is the maximum load if $m = n$?

- Using the Chernoff Bound:

$\mathbf{P}[\, X \geq t \,] \leq e^{-\mu}(e\mu/t)^t$

$$\mathbf{P}[\, X \geq t \,] \leq e^{-1}\left(\frac{e}{t}\right)^t \leq \left(\frac{e}{t}\right)^t$$

- By setting $t = 4\log n / \log\log n$, we claim to obtain $\mathbf{P}[\, X \geq t \,] \leq n^{-2}$.
- Indeed:

$$\left(\frac{e\log\log n}{4\log n}\right)^{4\log n/\log\log n} = \exp\left(\frac{4\log n}{\log\log n}\cdot\log\left(\frac{e\log\log n}{4\log n}\right)\right)$$

- The term inside the exponential is

$$\frac{4\log n}{\log\log n}\cdot\left(\log(4/e) + \log\log\log n - \log\log n\right) \leq \frac{4\log n}{\log\log n}\left(-\frac{1}{2}\log\log n\right),$$

obtaining that $\mathbf{P}[\, X \geq t \,] \leq n^{-4/2} = n^{-2}$.

This inequality only works for large enough n.

Balls into Bins: Bounding the Maximum Load (4/4)

We just proved that

$$\mathbf{P}[X \geq 4 \log n / \log \log n] \leq n^{-2},$$

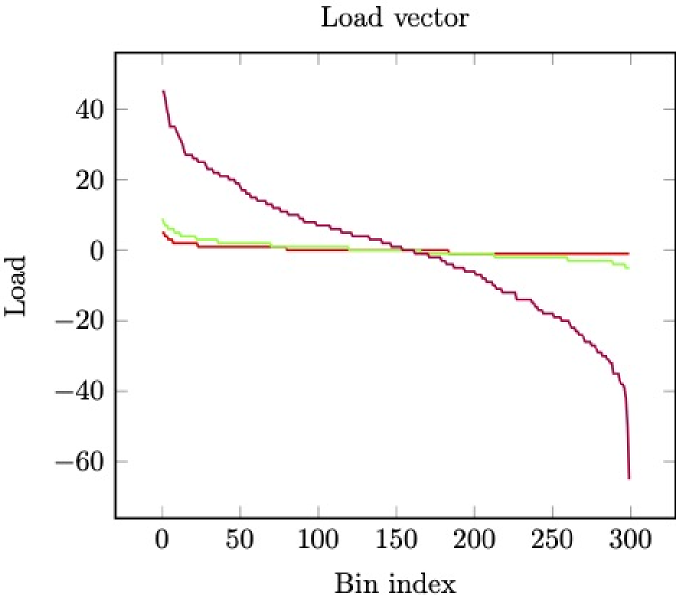
thus by the Union Bound, no bin receives more than $\Omega(\log n / \log \log n)$ balls with probability at least $1 - 1/n$. □

Simulations

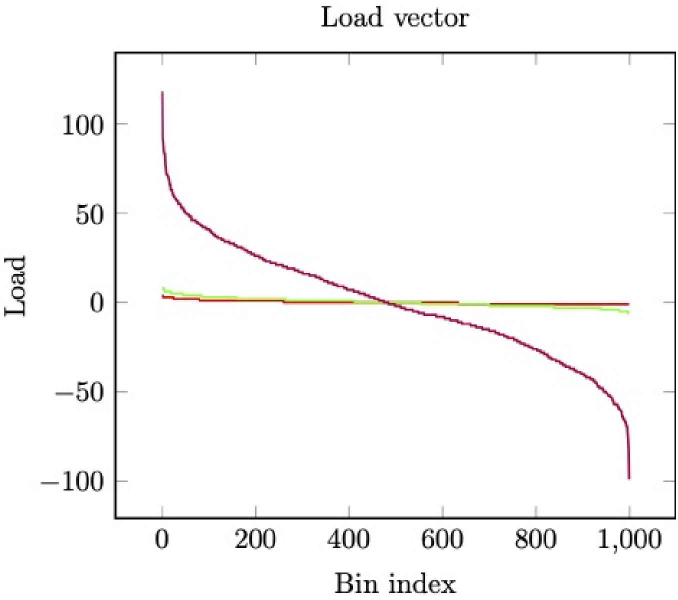
- We plot the load configuration for $m \in \{n, n \log n, n^2\}$
- We consider $n \in \{300, 1000, 100000\}$
- In plots, we take the normalised load, that is, actual bin load minus average load

Acknowledgements: experiments and plots created by Dimitris Los

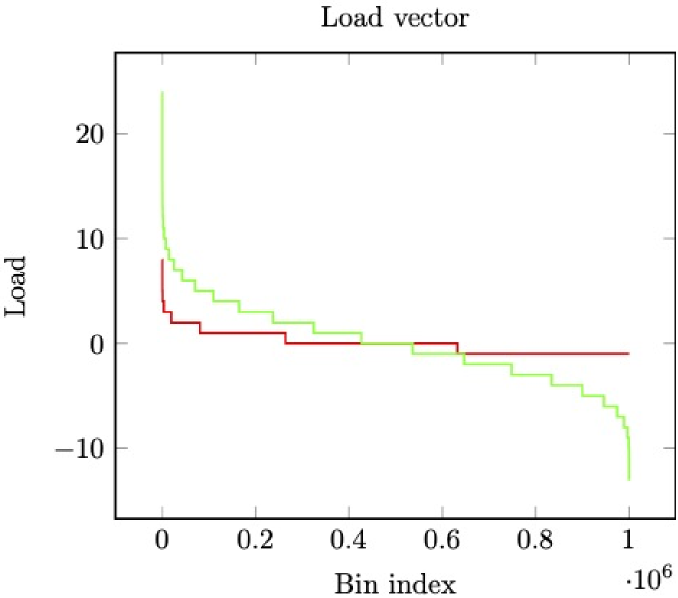
Balls-into-Bins Plot (1/3)



Balls-into-Bins Plot (2/3)



Balls-into-Bins Plot (3/3) (only $m \in \{n, n \log n\}$)



Conclusions

- If the number of balls is $2 \log n$ times n (the number of bins), then to distribute balls at random is a good algorithm

Conclusions

- If the number of balls is $2 \log n$ times n (the number of bins), then to distribute balls at random is a good algorithm
  - This is because the worst case maximum load is whp. $6 \log n$, while the average load is $2 \log n$

Conclusions

- If the number of balls is $2 \log n$ times n (the number of bins), then to distribute balls at random is a good algorithm
  - This is because the worst case maximum load is whp. $6 \log n$, while the average load is $2 \log n$
- For the case $m = n$, the algorithm is not good, since the maximum load is whp. $\Theta(\log n / \log \log n)$, while the average load is 1.

Conclusions

- If the number of balls is $2\log n$ times n (the number of bins), then to distribute balls at random is a good algorithm
  - This is because the worst case maximum load is whp. $6\log n$, while the average load is $2\log n$
- For the case $m = n$, the algorithm is not good, since the maximum load is whp. $\Theta(\log n / \log\log n)$, while the average load is 1.

A Better Load Balancing Approach

For any $m \geq n$, we can improve the balls into bin process by sampling two bins in each step, then assigning the ball into the bin with lesser load.

Conclusions

- If the number of balls is $2 \log n$ times n (the number of bins), then to distribute balls at random is a good algorithm
  - This is because the worst case maximum load is whp. $6 \log n$, while the average load is $2 \log n$
- For the case $m = n$, the algorithm is not good, since the maximum load is whp. $\Theta(\log n / \log \log n)$, while the average load is 1.

A Better Load Balancing Approach

For any $m \geq n$, we can improve the balls into bin process by sampling two bins in each step, then assigning the ball into the bin with lesser load.
$\Rightarrow$ gives a (normalised) maximum load $\Theta(\log \log n)$ w.p. $1 - 1/n$.

Conclusions

- If the number of balls is $2\log n$ times n (the number of bins), then to distribute balls at random is a good algorithm
  - This is because the worst case maximum load is whp. $6\log n$, while the average load is $2\log n$
- For the case $m = n$, the algorithm is not good, since the maximum load is whp. $\Theta(\log n / \log\log n)$, while the average load is 1.

A Better Load Balancing Approach

For any $m \geq n$, we can improve the balls into bin process by sampling two bins in each step, then assigning the ball into the bin with lesser load.
$\Rightarrow$ gives a (normalised) maximum load $\Theta(\log\log n)$ w.p. $1 - 1/n$.

This is called the **power of two choices**: It is a common technique to improve the performance of randomised algorithms.

Outline

QuickSort

```
QUICKSORT (Input A[1], A[2], ..., A[n])
Pick an element from the array, the so-called pivot
If |A| = 0 or |A| = 1 then
    return A
else
    Create two subarrays A1 and A2 (without the pivot) such that:
        A1 contains the elements that are smaller than the pivot
        A2 contains the elements that are greater (or equal) than the pivot
    QUICKSORT(A1)
    QUICKSORT(A2)
    return A
```

QuickSort

QuickSort (Input $A[1], A[2], \ldots, A[n]$)
Pick an element from the array, the so-called pivot
If $|A| = 0$ or $|A| = 1$ **then**
    return A
else
    Create two subarrays A_1 and A_2 (without the pivot) such that:
        A_1 contains the elements that are smaller than the pivot
        A_2 contains the elements that are greater (or equal) than the pivot
    QuickSort(A_1)
    QuickSort(A_2)
    return A

- Example: Let $A = (2, 8, 9, 1, 7, 5, 6, 3, 4)$ with $A[7] = 6$ as pivot.

QuickSort

```
QUICKSORT (Input A[1], A[2], ..., A[n])
Pick an element from the array, the so-called pivot
If |A| = 0 or |A| = 1 then
    return A
else
    Create two subarrays A_1 and A_2 (without the pivot) such that:
        A_1 contains the elements that are smaller than the pivot
        A_2 contains the elements that are greater (or equal) than the pivot
    QUICKSORT(A_1)
    QUICKSORT(A_2)
    return A
```

- Example: Let $A = (2, 8, 9, 1, 7, 5, 6, 3, 4)$ with $A[7] = 6$ as pivot.
  $\Rightarrow A_1 = (2, 1, 5, 3, 4)$ and $A_2 = (8, 9, 7)$

QuickSort

QuickSort (Input $A[1], A[2], \ldots, A[n]$)

Pick an element from the array, the so-called pivot
If $|A| = 0$ or $|A| = 1$ **then**
    return A
else
    Create two subarrays A_1 and A_2 (without the pivot) such that:
        A_1 contains the elements that are smaller than the pivot
        A_2 contains the elements that are greater (or equal) than the pivot
    QuickSort(A_1)
    QuickSort(A_2)
    return A

- Example: Let $A = (2, 8, 9, 1, 7, 5, 6, 3, 4)$ with $A[7] = 6$ as pivot.
  $\Rightarrow A_1 = (2, 1, 5, 3, 4)$ and $A_2 = (8, 9, 7)$
- Worst-Case Complexity (number of comparisons) is $\Theta(n^2)$,

QuickSort

QuickSort (Input $A[1], A[2], \ldots, A[n]$)

Pick an element from the array, the so-called pivot
If $|A| = 0$ or $|A| = 1$ **then**
    return A
else
    Create two subarrays A_1 and A_2 (without the pivot) such that:
        A_1 contains the elements that are smaller than the pivot
        A_2 contains the elements that are greater (or equal) than the pivot
    QuickSort(A_1)
    QuickSort(A_2)
    return A

- Example: Let $A = (2, 8, 9, 1, 7, 5, 6, 3, 4)$ with $A[7] = 6$ as pivot.
  $\Rightarrow A_1 = (2, 1, 5, 3, 4)$ and $A_2 = (8, 9, 7)$
- Worst-Case Complexity (number of comparisons) is $\Theta(n^2)$,
  while Average-Case Complexity is $O(n \log n)$.

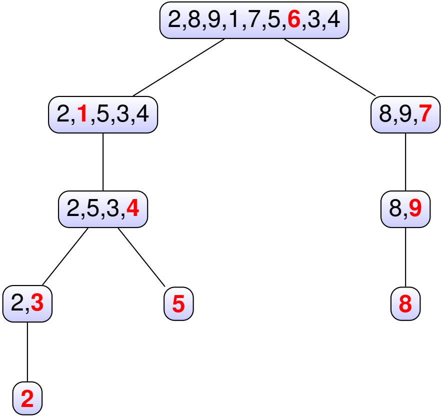
QuickSort

```
QUICKSORT (Input A[1], A[2], ..., A[n])
Pick an element from the array, the so-called pivot
If |A| = 0 or |A| = 1 then
    return A
else
    Create two subarrays A_1 and A_2 (without the pivot) such that:
        A_1 contains the elements that are smaller than the pivot
        A_2 contains the elements that are greater (or equal) than the pivot
    QUICKSORT(A_1)
    QUICKSORT(A_2)
    return A
```

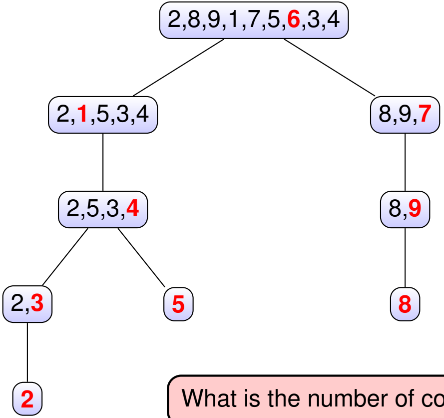
- Example: Let $A = (2, 8, 9, 1, 7, 5, 6, 3, 4)$ with $A[7] = 6$ as pivot.
  $\Rightarrow A_1 = (2, 1, 5, 3, 4)$ and $A_2 = (8, 9, 7)$
- Worst-Case Complexity (number of comparisons) is $\Theta(n^2)$, while Average-Case Complexity is $O(n \log n)$.

We will now give a proof of this “well-known” result!

QuickSort: How to Count Comparisons

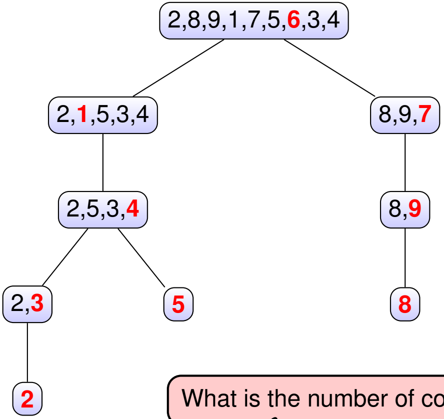


QuickSort: How to Count Comparisons



What is the number of comparisons?

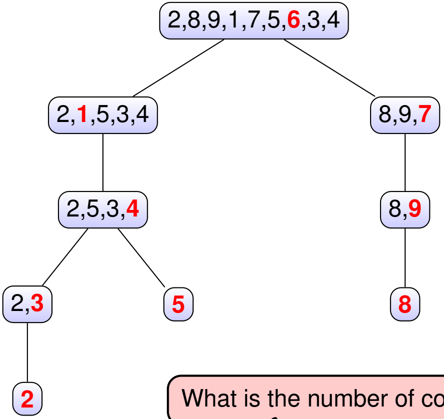
QuickSort: How to Count Comparisons



What is the number of comparisons?

Note that the number of comparison by QUICKSORT is equivalent to the sum of the height of all nodes in the tree (why?).

QuickSort: How to Count Comparisons



What is the number of comparisons?

Note that the number of comparison by QUICKSORT is equivalent to the sum of the height of all nodes in the tree (why?). In this case:

$$0 + 1 + 1 + 2 + 2 + 3 + 3 + 3 + 4 = 19.$$

Randomised QuickSort: Analysis (1/4)

How to pick a good pivot? We don't, **just pick one at random.**

Randomised QuickSort: Analysis (1/4)

How to pick a good pivot? We don't, **just pick one at random.**

This should be your standard answer in this course ☺

Randomised QuickSort: Analysis (1/4)

How to pick a good pivot? We don't, **just pick one at random.**

This should be your standard answer in this course ☺

Let us analyse QUICKSORT with random pivots.

Randomised QuickSort: Analysis (1/4)

How to pick a good pivot? We don't, **just pick one at random.**

This should be your standard answer in this course ☺

Let us analyse QUICKSORT with random pivots.

1. Assume A consists of n different numbers, w.l.o.g., $\{1, 2, \ldots, n\}$

Randomised QuickSort: Analysis (1/4)

How to pick a good pivot? We don't, **just pick one at random.**

This should be your standard answer in this course ☺

Let us analyse QUICKSORT with random pivots.

1. Assume A consists of n different numbers, w.l.o.g., $\{1, 2, \ldots, n\}$
2. Let H_i be the deepest level where element i appears in the tree.
   Then the number of comparison is $H = \sum_{i=1}^{n} H_i$

Randomised QuickSort: Analysis (1/4)

How to pick a good pivot? We don't, **just pick one at random.**

This should be your standard answer in this course ☺

Let us analyse QUICKSORT with random pivots.

1. Assume A consists of n different numbers, w.l.o.g., $\{1, 2, \ldots, n\}$
2. Let H_i be the deepest level where element i appears in the tree. Then the number of comparison is $H = \sum_{i=1}^{n} H_i$
3. We will prove that exists $C > 0$ such that

$$\mathbf{P}[H \leq Cn \log n] \geq 1 - n^{-1}.$$

Randomised QuickSort: Analysis (1/4)

How to pick a good pivot? We don't, **just pick one at random.**

This should be your standard answer in this course ☺

Let us analyse QUICKSORT with random pivots.

1. Assume A consists of n different numbers, w.l.o.g., $\{1, 2, \ldots, n\}$
2. Let H_i be the deepest level where element i appears in the tree.
   Then the number of comparison is $H = \sum_{i=1}^{n} H_i$
3. We will prove that exists $C > 0$ such that

$$\mathbf{P}\left[H \leq Cn \log n \right] \geq 1 - n^{-1}.$$

4. Actually, we will prove sth slightly stronger:

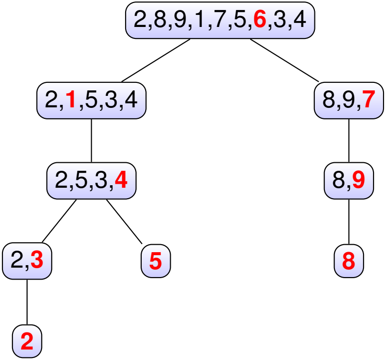
$$\mathbf{P}\left[\bigcap_{i=1}^{n} \{ H_i \leq C \log n \} \right] \geq 1 - n^{-1}.$$

Randomised QuickSort: Analysis (2/4)

- Let P be a path from the root to the deepest level of some element

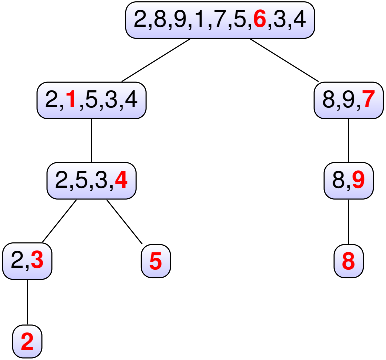
Randomised QuickSort: Analysis (2/4)

- Let P be a path from the root to the deepest level of some element



Randomised QuickSort: Analysis (2/4)

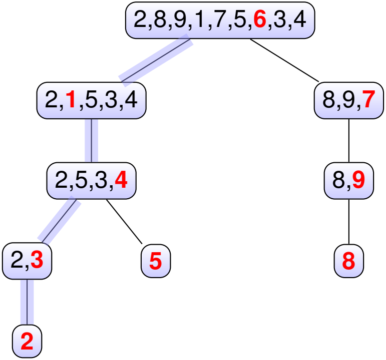
- Let P be a path from the root to the deepest level of some element



- Element 2: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (2, \mathbf{1}, 5, 3, 4) \to (2, 5, 3, \mathbf{4}) \to (2, \mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

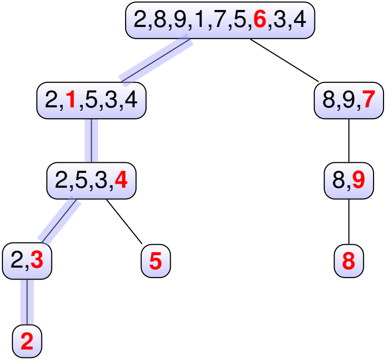
- Let P be a path from the root to the deepest level of some element



- Element 2: $(2,8,9,1,7,5,\mathbf{6},3,4) \to (2,\mathbf{1},5,3,4) \to (2,5,3,\mathbf{4}) \to (2,\mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

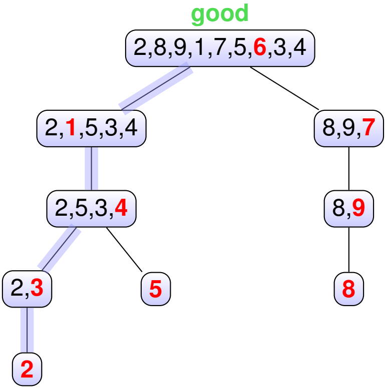
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**



- Element 2: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (2, \mathbf{1}, 5, 3, 4) \to (2, 5, 3, \mathbf{4}) \to (2, \mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

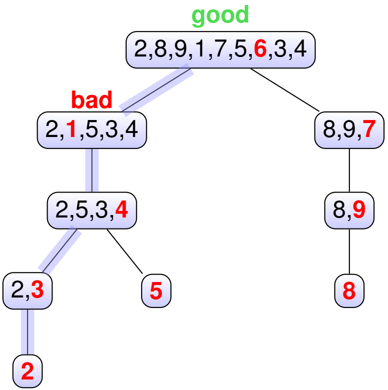
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**



- Element 2: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (2, \mathbf{1}, 5, 3, 4) \to (2, 5, 3, \mathbf{4}) \to (2, \mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

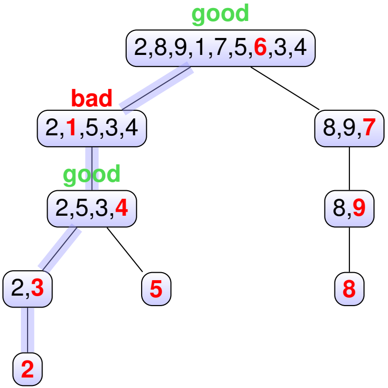
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**



- Element 2: $(2,8,9,1,7,5,\mathbf{6},3,4) \to (2,\mathbf{1},5,3,4) \to (2,5,3,\mathbf{4}) \to (2,\mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

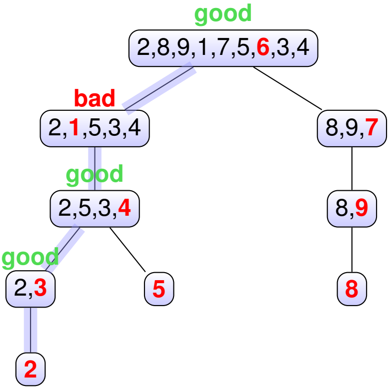
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**



- Element 2: $(2,8,9,1,7,5,\mathbf{6},3,4) \to (2,\mathbf{1},5,3,4) \to (2,5,3,\mathbf{4}) \to (2,\mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

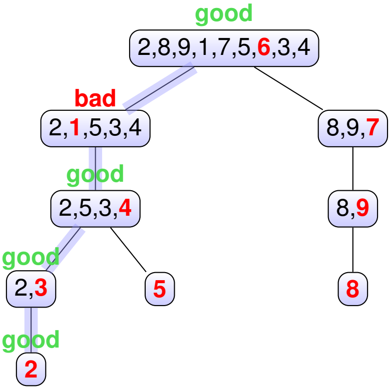
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**



- Element 2: $(2,8,9,1,7,5,\mathbf{6},3,4) \rightarrow (2,\mathbf{1},5,3,4) \rightarrow (2,5,3,\mathbf{4}) \rightarrow (2,\mathbf{3}) \rightarrow (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

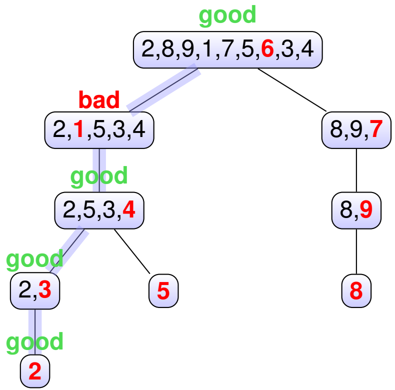
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**



- Element 2: $(2,8,9,1,7,5,\mathbf{6},3,4) \rightarrow (2,\mathbf{1},5,3,4) \rightarrow (2,5,3,\mathbf{4}) \rightarrow (2,\mathbf{3}) \rightarrow (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

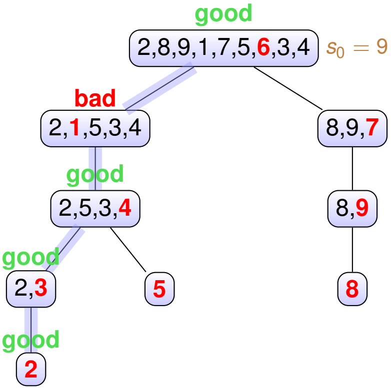
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**
- Further let s_t be the size of the array at level t in P.



- Element 2: $(2,8,9,1,7,5,\mathbf{6},3,4) \to (2,\mathbf{1},5,3,4) \to (2,5,3,\mathbf{4}) \to (2,\mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

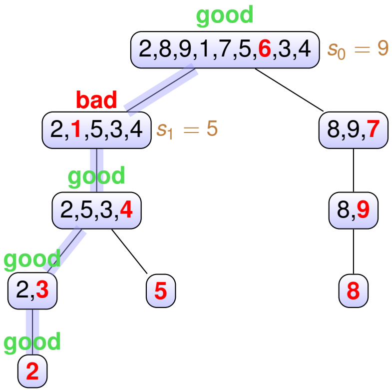
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**
- Further let s_t be the size of the array at level t in P.



- Element 2: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (2, \mathbf{1}, 5, 3, 4) \to (2, 5, 3, \mathbf{4}) \to (2, \mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

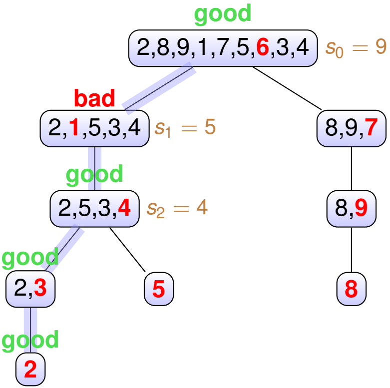
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**
- Further let s_t be the size of the array at level t in P.



- Element 2: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (2, \mathbf{1}, 5, 3, 4) \to (2, 5, 3, \mathbf{4}) \to (2, \mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

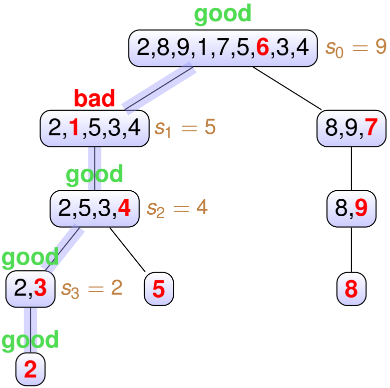
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**
- Further let s_t be the size of the array at level t in P.



- Element 2: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (2, \mathbf{1}, 5, 3, 4) \to (2, 5, 3, \mathbf{4}) \to (2, \mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

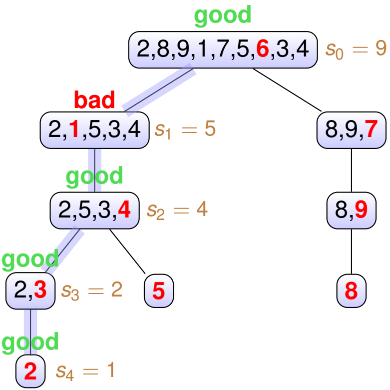
- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**
- Further let s_t be the size of the array at level t in P.



- Element 2: $(2,8,9,1,7,5,\mathbf{6},3,4) \to (2,\mathbf{1},5,3,4) \to (2,5,3,\mathbf{4}) \to (2,\mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (2/4)

- Let P be a path from the root to the deepest level of some element
  - A node in P is called **good** if the corresponding pivot partitions the array into two subarrays each of size at most $2/3$ of the previous one
  - otherwise, the node is **bad**
- Further let s_t be the size of the array at level t in P.



- Element 2: $(2, 8, 9, 1, 7, 5, \mathbf{6}, 3, 4) \to (2, \mathbf{1}, 5, 3, 4) \to (2, 5, 3, \mathbf{4}) \to (2, \mathbf{3}) \to (\mathbf{2})$

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k + 1$, the condition $s_k > 1$ is necessary

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k+1$, the condition $s_k > 1$ is necessary

How far could such a path P possibly run until we have $s_k = 1$?

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k + 1$, the condition $s_k > 1$ is necessary

> How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k+1$, the condition $s_k > 1$ is necessary

> How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$
- First Case, **good** node: $s_{k+1} \leq \frac{2}{3} \cdot s_k$.

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k+1$, the condition $s_k > 1$ is necessary

> How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$
- First Case, **good** node: $s_{k+1} \leq \frac{2}{3} \cdot s_k$.
- Second Case, **bad** node: $s_{k+1} \leq s_k$.

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k + 1$, the condition $s_k > 1$ is necessary

> How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$
- First Case, **good** node: $s_{k+1} \leq \frac{2}{3} \cdot s_k$.
- Second Case, **bad** node: $s_{k+1} \leq s_k$.

$\Rightarrow$ There are at most $T = \frac{\log n}{\log(3/2)} < 3 \log n$ many **good** nodes on any path P.

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k+1$, the condition $s_k > 1$ is necessary

> How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$
- First Case, **good** node: $s_{k+1} \leq \frac{2}{3} \cdot s_k$.
- Second Case, **bad** node: $s_{k+1} \leq s_k$.

> This even holds always, i.e., deterministically!

$\Rightarrow$ There are at most $T = \frac{\log n}{\log(3/2)} < 3 \log n$ many **good** nodes on any path P.

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \dots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k+1$, the condition $s_k > 1$ is necessary

How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$
- First Case, **good** node: $s_{k+1} \leq \frac{2}{3} \cdot s_k$.
- Second Case, **bad** node: $s_{k+1} \leq s_k$.

This even holds always, i.e., deterministically!

$\Rightarrow$ There are at most $T = \frac{\log n}{\log(3/2)} < 3 \log n$ many **good** nodes on any path P.

- Assume $|P| \geq C \log n$ for $C := 24$

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k+1$, the condition $s_k > 1$ is necessary

How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$
- First Case, **good** node: $s_{k+1} \leq \frac{2}{3} \cdot s_k$.
- Second Case, **bad** node: $s_{k+1} \leq s_k$.

This even holds always, i.e., deterministically!

$\Rightarrow$ There are at most $T = \frac{\log n}{\log(3/2)} < 3 \log n$ many **good** nodes on any path P.

- Assume $|P| \geq C \log n$ for $C := 24$

  $\Rightarrow$ number of **bad** vertices in the first $24 \log n$ levels is more than $21 \log n$.

Randomised QuickSort: Analysis (3/4)

- Consider now any element $i \in \{1, 2, \ldots, n\}$ and construct the path $P = P(i)$ one level by one
- For P to proceed from level k to $k+1$, the condition $s_k > 1$ is necessary

How far could such a path P possibly run until we have $s_k = 1$?

- We start with $s_0 = n$
- First Case, **good** node: $s_{k+1} \leq \frac{2}{3} \cdot s_k$.
- Second Case, **bad** node: $s_{k+1} \leq s_k$.

This even holds always, i.e., deterministically!

$\Rightarrow$ There are at most $T = \frac{\log n}{\log(3/2)} < 3 \log n$ many **good** nodes on any path P.

- Assume $|P| \geq C \log n$ for $C := 24$

  $\Rightarrow$ number of **bad** vertices in the first $24 \log n$ levels is more than $21 \log n$.

Let us now upper bound the probability that this "bad event" happens!

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:

Randomised QuickSort: Analysis (4/4)

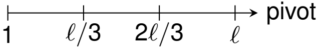
- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.
- $\mathbf{P}\left[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \right] \leq \frac{2}{3}$

Randomised QuickSort: Analysis (4/4)

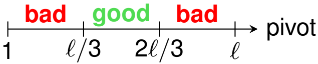
- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}\left[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,\right] \leq \frac{2}{3}$

Randomised QuickSort: Analysis (4/4)

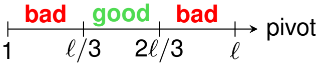
- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,] \leq \frac{2}{3}$

Randomised QuickSort: Analysis (4/4)

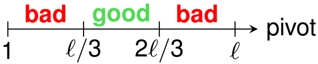
- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.

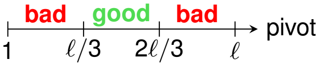


- $\mathbf{P}\left[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,\right] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

Question: But what if the path P does not reach level j?

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



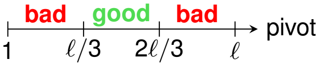
- $\mathbf{P}\left[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,\right] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

Question: But what if the path P does not reach level j?

Answer: We can then simply define X_j as the result of an independent coin flip with probability $2/3$.

Randomised QuickSort: Analysis (4/4)

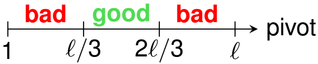
- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}\left[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \right] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.

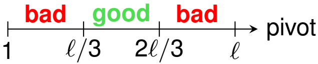


- $\mathbf{P}\left[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,\right] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the "nicer" **Chernoff Bound**!

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



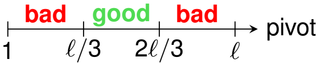
- $\mathbf{P}[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

> We can now apply the “nicer” **Chernoff Bound**!

- We have $\mathbf{E}[\, X \,] \leq (2/3) \cdot 24 \log n = 16 \log n$

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



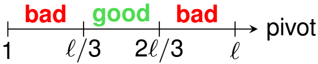
- $\mathbf{P}[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1}] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the "nicer" **Chernoff Bound**!

- We have $\mathbf{E}[X] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

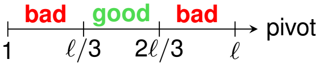
We can now apply the "nicer" **Chernoff Bound**!

- We have $\mathbf{E}[\, X \,] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

$$\mathbf{P}[\, X \geq \mathbf{E}[\, X \,] + t \,] \leq e^{-2t^2/n}$$

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1}] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the "nicer" **Chernoff Bound**!

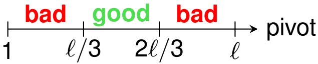
- We have $\mathbf{E}[X] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$

$$\mathbf{P}[X > 21 \log n] \leq \mathbf{P}[X > \mathbf{E}[X] + 5 \log n]$$

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1}] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the "nicer" **Chernoff Bound**!

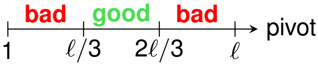
- We have $\mathbf{E}[X] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$

$$\mathbf{P}[X > 21 \log n] \leq \mathbf{P}[X > \mathbf{E}[X] + 5 \log n] \leq e^{-2(5 \log n)^2/(24 \log n)}$$

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1}] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the "nicer" **Chernoff Bound**!

- We have $\mathbf{E}[X] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

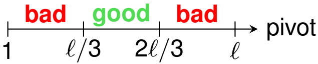
$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$

$$\mathbf{P}[X > 21 \log n] \leq \mathbf{P}[X > \mathbf{E}[X] + 5 \log n] \leq e^{-2(5 \log n)^2/(24 \log n)}$$

$$= e^{-(50/24) \log n}$$

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1}] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the "nicer" **Chernoff Bound**!

- We have $\mathbf{E}[X] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

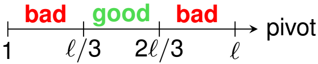
$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$

$$\mathbf{P}[X > 21 \log n] \leq \mathbf{P}[X > \mathbf{E}[X] + 5 \log n] \leq e^{-2(5 \log n)^2/(24 \log n)}$$

$$= e^{-(50/24) \log n} \leq n^{-2}.$$

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1}] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

> We can now apply the "nicer" **Chernoff Bound**!

- We have $\mathbf{E}[X] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

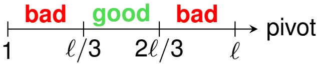
$\mathbf{P}[X \geq \mathbf{E}[X] + t] \leq e^{-2t^2/n}$

$$\mathbf{P}[X > 21 \log n] \leq \mathbf{P}[X > \mathbf{E}[X] + 5 \log n] \leq e^{-2(5 \log n)^2/(24 \log n)}$$
$$= e^{-(50/24) \log n} \leq n^{-2}.$$

- Hence P has more than $24 \log n$ nodes with probability at most n^{-2}.

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the “nicer” **Chernoff Bound**!

- We have $\mathbf{E}[\, X \,] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the “nicer” Chernoff Bounds

$$\mathbf{P}[\, X \geq \mathbf{E}[\, X \,] + t \,] \leq e^{-2t^2/n}$$

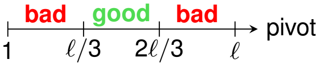
$$\mathbf{P}[\, X > 21 \log n \,] \leq \mathbf{P}[\, X > \mathbf{E}[\, X \,] + 5 \log n \,] \leq e^{-2(5 \log n)^2/(24 \log n)}$$

$$= e^{-(50/24) \log n} \leq n^{-2}.$$

- Hence P has more than $24 \log n$ nodes with probability at most n^{-2}.
- As there are in total n paths, by the union bound, the probability that at least one of them has more than $24 \log n$ nodes is at most n^{-1}.

Randomised QuickSort: Analysis (4/4)

- Consider the first $24 \log n$ vertices of P to the deepest level of element i.
- For any level $j \in \{0, 1, \ldots, 24 \log n - 1\}$, define an indicator variable X_j:
  - $X_j = 1$ if the node at level j is **bad**
  - $X_j = 0$ if the node at level j is **good**.



- $\mathbf{P}[\, X_j = 1 \mid X_0 = x_0, \ldots, X_{j-1} = x_{j-1} \,] \leq \frac{2}{3}$
- $X := \sum_{j=0}^{24 \log n - 1} X_j$ satisfies relaxed independence assumption (slide 16)

We can now apply the "nicer" **Chernoff Bound**!

- We have $\mathbf{E}[\, X \,] \leq (2/3) \cdot 24 \log n = 16 \log n$
- Then, by the "nicer" Chernoff Bounds

$$\mathbf{P}[\, X \geq \mathbf{E}[\, X \,] + t \,] \leq e^{-2t^2/n}$$

$$\mathbf{P}[\, X > 21 \log n \,] \leq \mathbf{P}[\, X > \mathbf{E}[\, X \,] + 5 \log n \,] \leq e^{-2(5 \log n)^2/(24 \log n)}$$

$$= e^{-(50/24) \log n} \leq n^{-2}.$$

- Hence P has more than $24 \log n$ nodes with probability at most n^{-2}.
- As there are in total n paths, by the union bound, the probability that at least one of them has more than $24 \log n$ nodes is at most n^{-1}. $\square$

Randomised QuickSort: Final Remarks

- Well-known: any comparison-based sorting algorithm needs $\Omega(n \log n)$

Randomised QuickSort: Final Remarks

- Well-known: any comparison-based sorting algorithm needs $\Omega(n \log n)$
- A classical result: expected number of comparison of randomised QuickSort is $2n \log n + O(n)$ (see, e.g., book by Mitzenmacher & Upfal)

Randomised QuickSort: Final Remarks

- Well-known: any comparison-based sorting algorithm needs $\Omega(n \log n)$
- A classical result: expected number of comparison of randomised QuickSort is $2n \log n + O(n)$ (see, e.g., book by Mitzenmacher & Upfal)

Supervision Exercise: Our upper bound of $O(n \log n)$ whp also immediately implies a $O(n \log n)$ bound on the expected number of comparisons!

Randomised QuickSort: Final Remarks

- Well-known: any comparison-based sorting algorithm needs $\Omega(n \log n)$
- A classical result: expected number of comparison of randomised QuickSort is $2n \log n + O(n)$ (see, e.g., book by Mitzenmacher & Upfal)

> Supervision Exercise: Our upper bound of $O(n \log n)$ whp also immediately implies a $O(n \log n)$ bound on the expected number of comparisons!

- It is possible to deterministically find the best pivot element that divides the array into two subarrays of the same size.
- The latter requires to compute the median of the array in linear time, which is not easy...
- The presented randomised algorithm for QuickSort is much easier to implement!

Outline

Hoeffding's Extension

- Besides **sums of independent bernoulli** random variables, sums of independent and bounded random variables are very frequent in applications.

Hoeffding's Extension

- Besides **sums of independent bernoulli** random variables, sums of independent and bounded random variables are very frequent in applications.
- Unfortunately the distribution of the X_i may be unknown or hard to compute, thus it will be hard to compute the moment-generating function.

Hoeffding's Extension

- Besides **sums of independent bernoulli** random variables, sums of independent and bounded random variables are very frequent in applications.
- Unfortunately the distribution of the X_i may be unknown or hard to compute, thus it will be hard to compute the moment-generating function.
- Hoeffding's Lemma helps us here:

Hoeffding's Extension

- Besides **sums of independent bernoulli** random variables, sums of independent and bounded random variables are very frequent in applications.
- Unfortunately the distribution of the X_i may be unknown or hard to compute, thus it will be hard to compute the moment-generating function.
- Hoeffding's Lemma helps us here:

Hoeffding's Extension Lemma

Let X be a random variable with mean 0 such that $a \leq X \leq b$. Then for all $\lambda \in \mathbb{R}$,

$$\mathbf{E}\left[e^{\lambda X}\right] \leq \exp\left(\frac{(b-a)^2\lambda^2}{8}\right)$$

Hoeffding's Extension

- Besides **sums of independent bernoulli** random variables, sums of independent and bounded random variables are very frequent in applications.
- Unfortunately the distribution of the X_i may be unknown or hard to compute, thus it will be hard to compute the moment-generating function.
- Hoeffding's Lemma helps us here:

You can always consider $X' = X - \mathbf{E}[X]$

Hoeffding's Extension Lemma

Let X be a random variable with mean 0 such that $a \leq X \leq b$. Then for all $\lambda \in \mathbb{R}$,

$$\mathbf{E}\left[e^{\lambda X}\right] \leq \exp\left(\frac{(b-a)^2\lambda^2}{8}\right)$$

Hoeffding's Extension

- Besides **sums of independent bernoulli** random variables, sums of independent and bounded random variables are very frequent in applications.
- Unfortunately the distribution of the X_i may be unknown or hard to compute, thus it will be hard to compute the moment-generating function.
- Hoeffding's Lemma helps us here:

You can always consider $X' = X - \mathbf{E}[X]$

Hoeffding's Extension Lemma

Let X be a random variable with mean 0 such that $a \leq X \leq b$. Then for all $\lambda \in \mathbb{R}$,

$$\mathbf{E}\left[e^{\lambda X}\right] \leq \exp\left(\frac{(b-a)^2\lambda^2}{8}\right)$$

We omit the proof of this lemma!

Hoeffding Bounds

Hoeffding's Inequality

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[X] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$

and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Hoeffding Bounds

Hoeffding's Inequality

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[X] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$

and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Proof Outline (skipped):

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots + X_n'$, then $\mathbf{P}[X \geq \mu + t] = \mathbf{P}[X' \geq t]$

Hoeffding Bounds

Hoeffding's Inequality

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[X] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$

and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Proof Outline (skipped):

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots + X_n'$, then $\mathbf{P}[X \geq \mu + t] = \mathbf{P}[X' \geq t]$
- $\mathbf{P}[X' \geq t] \leq e^{-\lambda t} \prod_{i=1}^n \mathbf{E}\left[e^{\lambda X_i'}\right] \leq \exp\left[-\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^n (b_i - a_i)^2\right]$

Hoeffding Bounds

Hoeffding's Inequality

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[X] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$

and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Proof Outline (skipped):

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots + X_n'$, then $\mathbf{P}[X \geq \mu + t] = \mathbf{P}[X' \geq t]$
- $\mathbf{P}[X' \geq t] \leq e^{-\lambda t} \prod_{i=1}^n \mathbf{E}\left[e^{\lambda X_i'}\right] \leq \exp\left[-\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^n (b_i - a_i)^2\right]$
- Choose $\lambda = \frac{4t}{\sum_{i=1}^n (b_i - a_i)^2}$ to get the result.

Hoeffding Bounds

Hoeffding's Inequality

Let $X_1, \ldots, X_n$ be independent random variable with mean μ_i such that $a_i \leq X_i \leq b_i$. Let $X = X_1 + \ldots + X_n$, and let $\mu = \mathbf{E}[X] = \sum_{i=1}^n \mu_i$. Then for any $t > 0$

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$

and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Proof Outline (skipped):

- Let $X_i' = X_i - \mu_i$ and $X' = X_1' + \ldots + X_n'$, then $\mathbf{P}[X \geq \mu + t] = \mathbf{P}[X' \geq t]$
- $\mathbf{P}[X' \geq t] \leq e^{-\lambda t} \prod_{i=1}^n \mathbf{E}\left[e^{\lambda X_i'}\right] \leq \exp\left[-\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^n (b_i - a_i)^2\right]$
- Choose $\lambda = \frac{4t}{\sum_{i=1}^n (b_i - a_i)^2}$ to get the result.

This is not magic! you just need to optimise λ!

Method of Bounded Differences

Framework

Suppose, we have independent random variables $X_1, \ldots, X_n$. We want to study the random variable:

$$f(X_1, \ldots, X_n)$$

Method of Bounded Differences

Framework

Suppose, we have independent random variables $X_1, \ldots, X_n$. We want to study the random variable:

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$

Method of Bounded Differences

Framework

Suppose, we have independent random variables $X_1, \ldots, X_n$. We want to study the random variable:

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$
2. In balls into bins, X_i indicates where ball i is allocated, and $f(X_1, \ldots, X_m)$ is the number of empty bins

Method of Bounded Differences

Framework

Suppose, we have independent random variables $X_1, \ldots, X_n$. We want to study the random variable:

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$
2. In balls into bins, X_i indicates where ball i is allocated, and $f(X_1, \ldots, X_m)$ is the number of empty bins
3. X_i indicates if the i-th edge is present in a graph, and $f(X_1, \ldots, X_m)$ represents the number of connected components of G

Method of Bounded Differences

Framework

Suppose, we have independent random variables $X_1, \ldots, X_n$. We want to study the random variable:

$$f(X_1, \ldots, X_n)$$

Some examples:

1. $X = X_1 + \ldots + X_n$
2. In balls into bins, X_i indicates where ball i is allocated, and $f(X_1, \ldots, X_m)$ is the number of empty bins
3. X_i indicates if the i-th edge is present in a graph, and $f(X_1, \ldots, X_m)$ represents the number of connected components of G

In all those cases (and more) we can easily prove concentration of $f(X_1, \ldots, X_n)$ around its mean by the so-called **Method of Bounded Differences**.

Method of Bounded Differences

A function f is called Lipschitz with parameters $\mathbf{c} = (c_1, \ldots, c_n)$ if for all $i = 1, 2, \ldots, n$,

$$|f(x_1, x_2, \ldots, x_{i-1}, \boldsymbol{x_i}, x_{i+1}, \ldots, x_n) - f(x_1, x_2, \ldots, x_{i-1}, \boldsymbol{y_i}, x_{i+1}, \ldots, x_n)| \leq c_i,$$

where x_i and y_i are in the domain of the i-th coordinate.

Method of Bounded Differences

A function f is called Lipschitz with parameters $\mathbf{c} = (c_1, \ldots, c_n)$ if for all $i = 1, 2, \ldots, n$,

$$|f(x_1, x_2, \ldots, x_{i-1}, \mathbf{x_i}, x_{i+1}, \ldots, x_n) - f(x_1, x_2, \ldots, x_{i-1}, \mathbf{y_i}, x_{i+1}, \ldots, x_n)| \leq c_i,$$

where x_i and y_i are in the domain of the i-th coordinate.

McDiarmid's inequality

Let $X_1, \ldots, X_n$ be independent random variables. Let f be Lipschitz with parameters $\mathbf{c} = (c_1, \ldots, c_n)$. Let $X = f(X_1, \ldots, X_n)$. Then for any $t > 0$,

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right),$$

and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right).$$

Method of Bounded Differences

A function f is called Lipschitz with parameters $\mathbf{c} = (c_1, \ldots, c_n)$ if for all $i = 1, 2, \ldots, n$,

$$|f(x_1, x_2, \ldots, x_{i-1}, \mathbf{x_i}, x_{i+1}, \ldots, x_n) - f(x_1, x_2, \ldots, x_{i-1}, \mathbf{y_i}, x_{i+1}, \ldots, x_n)| \leq c_i,$$

where x_i and y_i are in the domain of the i-th coordinate.

McDiarmid's inequality

Let $X_1, \ldots, X_n$ be independent random variables. Let f be Lipschitz with parameters $\mathbf{c} = (c_1, \ldots, c_n)$. Let $X = f(X_1, \ldots, X_n)$. Then for any $t > 0$,

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right),$$

and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right).$$

- Notice the similarity with Hoeffding's inequality!

Method of Bounded Differences

A function f is called Lipschitz with parameters $\mathbf{c} = (c_1, \ldots, c_n)$ if for all $i = 1, 2, \ldots, n$,

$$|f(x_1, x_2, \ldots, x_{i-1}, \mathbf{x_i}, x_{i+1}, \ldots, x_n) - f(x_1, x_2, \ldots, x_{i-1}, \mathbf{y_i}, x_{i+1}, \ldots, x_n)| \leq c_i,$$

where x_i and y_i are in the domain of the i-th coordinate.

McDiarmid's inequality

Let $X_1, \ldots, X_n$ be independent random variables. Let f be Lipschitz with parameters $\mathbf{c} = (c_1, \ldots, c_n)$. Let $X = f(X_1, \ldots, X_n)$. Then for any $t > 0$,

$$\mathbf{P}[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right),$$

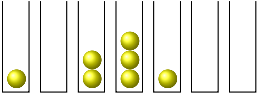
and

$$\mathbf{P}[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right).$$

- Notice the similarity with Hoeffding's inequality!
- The proof is omitted here (it requires the concept of martingales).

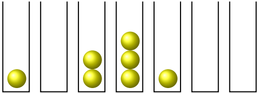
Outline

Application 3: Balls into Bins (again...)



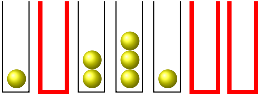
- Consider again m balls assigned uniformly at random into n bins.

Application 3: Balls into Bins (again...)



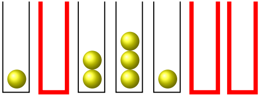
- Consider again m balls assigned uniformly at random into n bins.
- Enumerate the balls from 1 to m. Ball i is assigned to a random bin X_i

Application 3: Balls into Bins (again...)



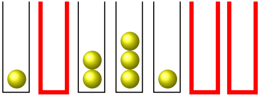
- Consider again m balls assigned uniformly at random into n bins.
- Enumerate the balls from 1 to m. Ball i is assigned to a random bin X_i
- Let Z be the number of empty bins (after assigning the m balls)

Application 3: Balls into Bins (again...)



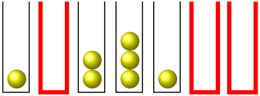
- Consider again m balls assigned uniformly at random into n bins.
- Enumerate the balls from 1 to m. Ball i is assigned to a random bin X_i
- Let Z be the number of empty bins (after assigning the m balls)
- $Z = Z(X_1, \ldots, X_m)$ and Z is Lipschitz with $\mathbf{c} = (1, \ldots, 1)$

Application 3: Balls into Bins (again...)



- Consider again m balls assigned uniformly at random into n bins.
- Enumerate the balls from 1 to m. Ball i is assigned to a random bin X_i
- Let Z be the number of empty bins (after assigning the m balls)
- $Z = Z(X_1, \ldots, X_m)$ and Z is Lipschitz with $\mathbf{c} = (1, \ldots, 1)$
  (If we move one ball to another bin, number of empty bins changes by ≤ 1.)

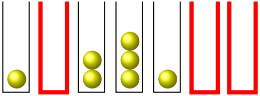
Application 3: Balls into Bins (again...)



- Consider again m balls assigned uniformly at random into n bins.
- Enumerate the balls from 1 to m. Ball i is assigned to a random bin X_i
- Let Z be the number of empty bins (after assigning the m balls)
- $Z = Z(X_1, \ldots, X_m)$ and Z is Lipschitz with $\mathbf{c} = (1, \ldots, 1)$
(If we move one ball to another bin, number of empty bins changes by ≤ 1.)
- By McDiarmid's inequality, for any $t \geq 0$,

$$\mathbf{P}\left[\,|Z - \mathbf{E}[Z]| > t\,\right] \leq 2 \cdot e^{-2t^2/m}.$$

Application 3: Balls into Bins (again...)

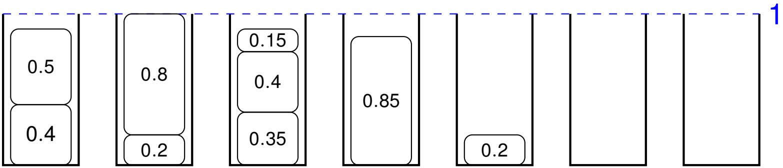


- Consider again m balls assigned uniformly at random into n bins.
- Enumerate the balls from 1 to m. Ball i is assigned to a random bin X_i
- Let Z be the number of empty bins (after assigning the m balls)
- $Z = Z(X_1, \ldots, X_m)$ and Z is Lipschitz with $\mathbf{c} = (1, \ldots, 1)$
  (If we move one ball to another bin, number of empty bins changes by ≤ 1.)
- By McDiarmid's inequality, for any $t \geq 0$,

$$\mathbf{P}\left[\,|Z - \mathbf{E}[Z]| > t\,\right] \leq 2 \cdot e^{-2t^2/m}.$$

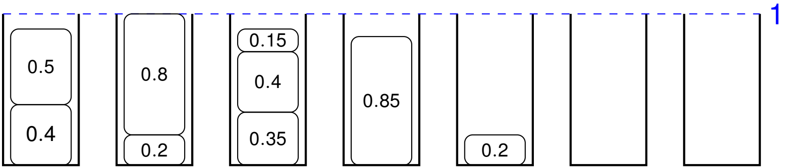
This is a decent bound, but for some values of m it is far from tight and stronger bounds are possible through a refined analysis.

Application 4: Bin Packing



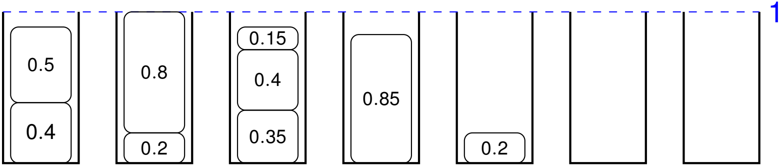
- We are given n items of sizes in the unit interval $[0, 1]$

Application 4: Bin Packing



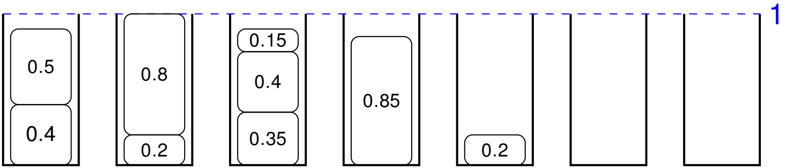
- We are given n items of sizes in the unit interval $[0, 1]$
- We want to pack those items into the fewest number of unit-capacity bins

Application 4: Bin Packing



- We are given n items of sizes in the unit interval $[0, 1]$
- We want to pack those items into the fewest number of unit-capacity bins
- Suppose the item sizes X_i are independent random variables in $[0, 1]$

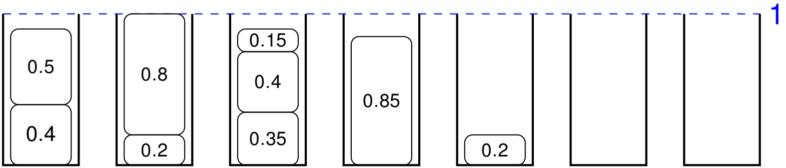
Application 4: Bin Packing



- We are given n items of sizes in the unit interval $[0, 1]$
- We want to pack those items into the fewest number of unit-capacity bins
- Suppose the item sizes X_i are independent random variables in $[0, 1]$

- Let $B = B(X_1, \ldots, X_n)$ be the optimal number of bins

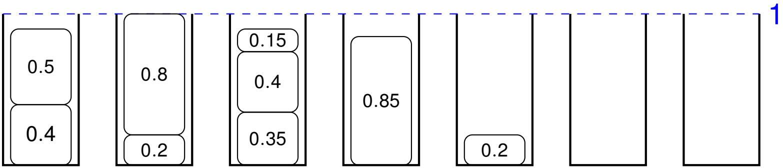
Application 4: Bin Packing



- We are given n items of sizes in the unit interval $[0, 1]$
- We want to pack those items into the fewest number of unit-capacity bins
- Suppose the item sizes X_i are independent random variables in $[0, 1]$

- Let $B = B(X_1, \dots, X_n)$ be the optimal number of bins
- The Lipschitz conditions holds with $\boldsymbol{c} = (1, \dots, 1)$. **Why?**

Application 4: Bin Packing

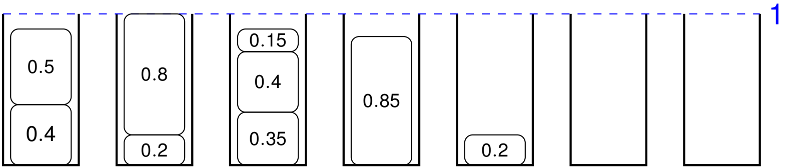


- We are given n items of sizes in the unit interval $[0, 1]$
- We want to pack those items into the fewest number of unit-capacity bins
- Suppose the item sizes X_i are independent random variables in $[0, 1]$

- Let $B = B(X_1, \ldots, X_n)$ be the optimal number of bins
- The Lipschitz conditions holds with $\boldsymbol{c} = (1, \ldots, 1)$. **Why?**
- Therefore

$$\mathbf{P}[\,|B - \mathbf{E}[B]| \geq t\,] \leq 2 \cdot e^{-2t^2/n}.$$

Application 4: Bin Packing



- We are given n items of sizes in the unit interval $[0, 1]$
- We want to pack those items into the fewest number of unit-capacity bins
- Suppose the item sizes X_i are independent random variables in $[0, 1]$

- Let $B = B(X_1, \ldots, X_n)$ be the optimal number of bins
- The Lipschitz conditions holds with $\boldsymbol{c} = (1, \ldots, 1)$. **Why?**
- Therefore

$$\mathbf{P}\left[\, |B - \mathbf{E}[\, B\,]| \geq t\,\right] \leq 2 \cdot e^{-2t^2/n}.$$

This is a typical example where proving concentration is much easier than calculating (or estimating) the expectation!

Outline

Moment Generating Functions

Moment-Generating Function

The moment-generating function of a random variable X is

$$M_X(t) = \mathbf{E}\left[e^{tX} \right], \qquad \text{where } t \in \mathbb{R}.$$

Moment Generating Functions

Moment-Generating Function

The moment-generating function of a random variable X is

$$M_X(t) = \mathbf{E}\left[e^{tX} \right], \qquad \text{where } t \in \mathbb{R}.$$

Using power series of e and differentiating shows that $M_X(t)$ encapsulates all moments of X.

Moment Generating Functions

Moment-Generating Function

The moment-generating function of a random variable X is

$$M_X(t) = \mathbf{E}\left[e^{tX}\right], \qquad \text{where } t \in \mathbb{R}.$$

Using power series of e and differentiating shows that $M_X(t)$ encapsulates all moments of X.

Lemma

1. If X and Y are two r.v.'s with $M_X(t) = M_Y(t)$ for all $t \in (-\delta, +\delta)$ for some $\delta > 0$, then the distributions X and Y are identical.
2. If X and Y are independent random variables, then

$$M_{X+Y}(t) = M_X(t) \cdot M_Y(t).$$

Moment Generating Functions

Moment-Generating Function

The moment-generating function of a random variable X is

$$M_X(t) = \mathbf{E}\left[e^{tX}\right], \qquad \text{where } t \in \mathbb{R}.$$

Using power series of e and differentiating shows that $M_X(t)$ encapsulates all moments of X.

Lemma

1. If X and Y are two r.v.'s with $M_X(t) = M_Y(t)$ for all $t \in (-\delta, +\delta)$ for some $\delta > 0$, then the distributions X and Y are identical.
2. If X and Y are independent random variables, then

$$M_{X+Y}(t) = M_X(t) \cdot M_Y(t).$$

Proof of 2:

$$M_{X+Y}(t) = \mathbf{E}\left[e^{t(X+Y)}\right] = \mathbf{E}\left[e^{tX} \cdot e^{tY}\right] \stackrel{(!)}{=} \mathbf{E}\left[e^{tX}\right] \cdot \mathbf{E}\left[e^{tY}\right] = M_X(t)M_Y(t) \qquad \square$$