Randomised Algorithms

Lecture 5: Random Walks, Hitting Times and Application to 2-SAT

Thomas Sauerwald (`tms41@cam.ac.uk`)

Lent 2022

UNIVERSITY OF CAMBRIDGE

Outline

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \not\in E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Random Walks on Graphs

A Simple Random Walk (SRW) on a graph G is a Markov chain on $V(G)$ with

$$P(u,v) = \begin{cases} \frac{1}{\deg(u)} & \text{if } \{u,v\} \in E, \\ 0 & \text{if } \{u,v\} \notin E. \end{cases}, \qquad \text{and} \qquad \pi(u) = \frac{\deg(u)}{2|E|}$$

Recall: $h(u,v) = \mathbf{E}_u[\min\{t \geq 1 \colon X_t = v\}]$ is the hitting time of v from u.

Lazy Random Walks and Periodicity

The Lazy Random Walk (LRW) on G given by $\widetilde{P} = (P + I)/2$,

$$\widetilde{P}_{u,v} = \begin{cases} \frac{1}{2\deg(u)} & \text{if } \{u,v\} \in E, \\ \frac{1}{2} & \text{if } u = v, \\ 0 & \text{otherwise} \end{cases} .$$

Lazy Random Walks and Periodicity

The Lazy Random Walk (LRW) on G given by $\widetilde{P} = (P + I)/2$,

$$\widetilde{P}_{u,v} = \begin{cases} \frac{1}{2\deg(u)} & \text{if } \{u, v\} \in E, \\ \frac{1}{2} & \text{if } u = v, \\ 0 & \text{otherwise} \end{cases}.$$

P - SRW matrix
I - Identity matrix.

Lazy Random Walks and Periodicity

The Lazy Random Walk (LRW) on G given by $\widetilde{P} = (P + I)/2$,

$$\widetilde{P}_{u,v} = \begin{cases} \frac{1}{2\deg(u)} & \text{if } \{u,v\} \in E, \\ \frac{1}{2} & \text{if } u = v, \\ 0 & \text{otherwise} \end{cases} .$$

P - SRW matrix
I - Identity matrix.

Fact: For any graph G the LRW on G is aperiodic.

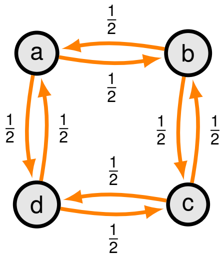
Lazy Random Walks and Periodicity

The Lazy Random Walk (LRW) on G given by $\widetilde{P} = (P + I)/2$,

$$\widetilde{P}_{u,v} = \begin{cases} \frac{1}{2\deg(u)} & \text{if } \{u,v\} \in E, \\ \frac{1}{2} & \text{if } u = v, \\ 0 & \text{otherwise} \end{cases} .$$

P - SRW matrix
I - Identity matrix.

Fact: For any graph G the LRW on G is aperiodic.



SRW on C_4, *Periodic*

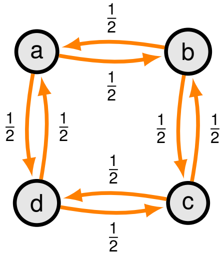
Lazy Random Walks and Periodicity

The Lazy Random Walk (LRW) on G given by $\widetilde{P} = (P + I)/2$,

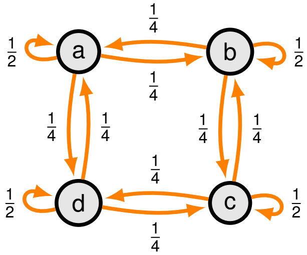
$$\widetilde{P}_{u,v} = \begin{cases} \frac{1}{2\deg(u)} & \text{if } \{u,v\} \in E, \\ \frac{1}{2} & \text{if } u = v, \\ 0 & \text{otherwise} \end{cases} .$$

P - SRW matrix
I - Identity matrix.

Fact: For any graph G the LRW on G is aperiodic.



SRW on C_4, *Periodic*



LRW on C_4, *Aperiodic*

Application: Cover Time and Undirected Connectivity

Let $t_{cov} := \max_{u \in V} \mathbf{E}_u\left[\min\{t \geq 1\colon \cup_{s=0}^{t} X_s = V\}\right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Application: Cover Time and Undirected Connectivity

Let $t_{cov} := \max_{u \in V} \mathbf{E}_u \left[\min\{ t \geq 1 \colon \cup_{s=0}^{t} X_s = V \} \right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

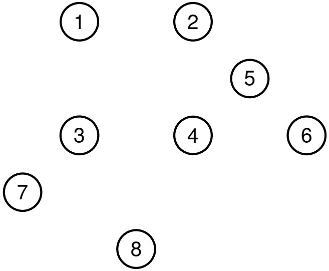
Application: Cover Time and Undirected Connectivity

Let $t_{cov} := \max_{u \in V} \mathbf{E}_u \left[\min\{ t \geq 1 : \cup_{s=0}^{t} X_s = V \} \right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

Proof:



Application: Cover Time and Undirected Connectivity

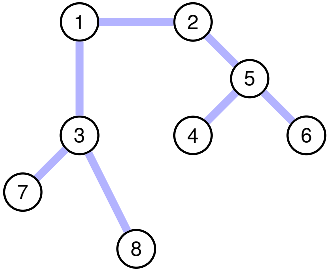
Let $t_{cov} := \max_{u \in V} \mathbf{E}_u\left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\}\right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

Proof:

- Take a spanning tree T in G



Application: Cover Time and Undirected Connectivity

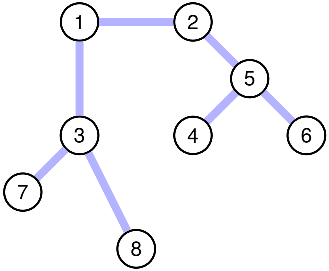
Let $t_{cov} := \max_{u \in V} \mathbf{E}_u \left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\} \right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

Proof:

- Take a spanning tree T in G
- Consider a traversal that goes through every edge in T twice



Application: Cover Time and Undirected Connectivity

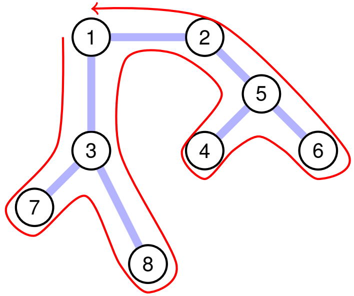
Let $t_{cov} := \max_{u \in V} \mathbf{E}_u\left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\}\right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

Proof:

- Take a spanning tree T in G
- Consider a traversal that goes through every edge in T twice



Application: Cover Time and Undirected Connectivity

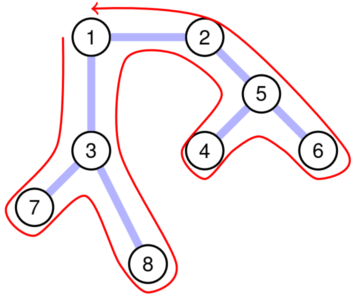
Let $t_{cov} := \max_{u \in V} \mathbf{E}_u\left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\}\right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

Proof:

- Take a spanning tree T in G
- Consider a traversal that goes through every edge in T twice
- For any adjacent vertices u, v, $t_{hit}(u,v) + t_{hit}(v,u) \leq 2|E|$ (Exercise!)



Application: Cover Time and Undirected Connectivity

Let $t_{cov} := \max_{u \in V} \mathbf{E}_u \left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\} \right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

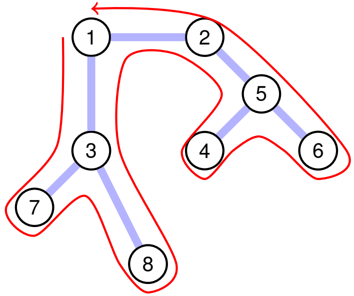
Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

Proof:

- Take a spanning tree T in G
- Consider a traversal that goes through every edge in T twice
- For any adjacent vertices u, v, $t_{hit}(u,v) + t_{hit}(v,u) \leq 2|E|$ (Exercise!)
- Thus,

$$t_{cov}(G) \leq \sum_{(u,v) \in E(T)} h(u,v) + h(v,u)$$



Application: Cover Time and Undirected Connectivity

Let $t_{cov} := \max_{u \in V} \mathbf{E}_u \left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\} \right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

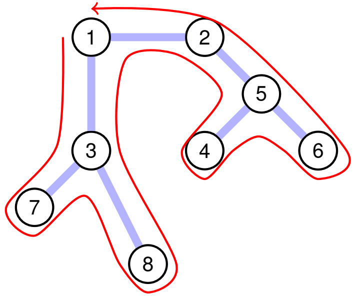
Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

Proof:

- Take a spanning tree T in G
- Consider a traversal that goes through every edge in T twice
- For any adjacent vertices u, v, $t_{hit}(u,v) + t_{hit}(v,u) \leq 2|E|$ (Exercise!)
- Thus,

$$t_{cov}(G) \leq \sum_{(u,v) \in E(T)} h(u,v) + h(v,u)$$
$$\leq 2(n-1) \cdot |E|.$$



Application: Cover Time and Undirected Connectivity

Let $t_{cov} := \max_{u \in V} \mathbf{E}_u \left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\} \right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

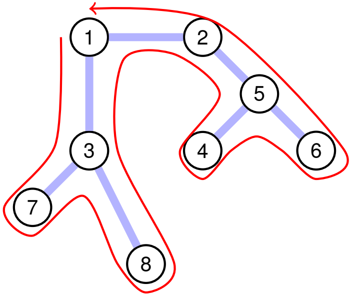
For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

By Markov's inequality, all vertices are visited after $4(n-1)|E|$ steps with probability at least $1/2$

Proof:

- Take a spanning tree T in G
- Consider a traversal that goes through every edge in T twice
- For any adjacent vertices u, v, $t_{hit}(u,v) + t_{hit}(v,u) \leq 2|E|$ (Exercise!)
- Thus,

$$t_{cov}(G) \leq \sum_{(u,v) \in E(T)} h(u,v) + h(v,u)$$
$$\leq 2(n-1) \cdot |E|.$$



Application: Cover Time and Undirected Connectivity

Let $t_{cov} := \max_{u \in V} \mathbf{E}_u\left[\min\{t \geq 1 \colon \cup_{s=0}^{t} X_s = V\}\right]$ be the cover time, that is, the worst-case expected time to visit all vertices.

Aleliunas, Karp, Lipton, Lovász and Rackoff, FOCS'79

For any connected graph G with n vertices, $t_{cov}(G) \leq 2(n-1)|E|$.

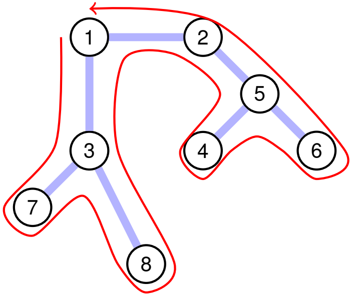
By Markov's inequality, all vertices are visited after $4(n-1)|E|$ steps with probability at least $1/2$

$\Rightarrow$ solves the USTCON problem with $O(\log n)$ space (see Complexity Theory course)

Proof:

- Take a spanning tree T in G
- Consider a traversal that goes through every edge in T twice
- For any adjacent vertices u, v, $t_{hit}(u,v) + t_{hit}(v,u) \leq 2|E|$ (Exercise!)
- Thus,

$$t_{cov}(G) \leq \sum_{(u,v) \in E(T)} h(u,v) + h(v,u)$$
$$\leq 2(n-1) \cdot |E|.$$



Outline

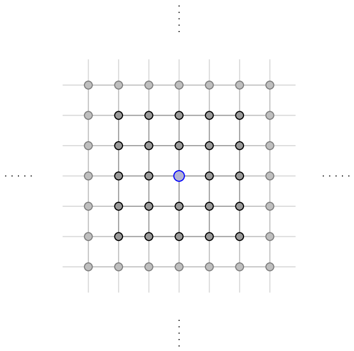
1921: The Birth of Random Walks on (Infinite) Graphs (Polyá)

Will a random walk always return to the origin?

1921: The Birth of Random Walks on (Infinite) Graphs (Polyá)

Will a random walk always return to the origin?

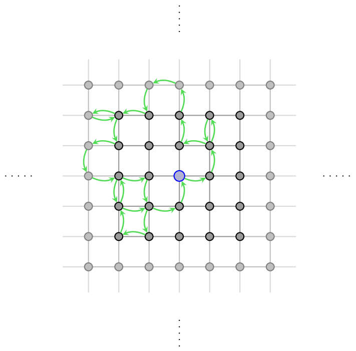
Infinite 2D Grid



1921: The Birth of Random Walks on (Infinite) Graphs (Polyá)

Will a random walk always return to the origin?

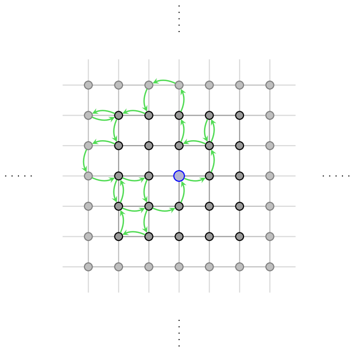
Infinite 2D Grid



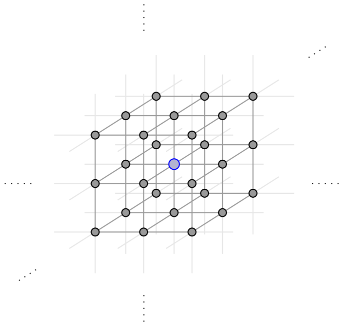
1921: The Birth of Random Walks on (Infinite) Graphs (Polyá)

Will a random walk always return to the origin?

Infinite 2D Grid



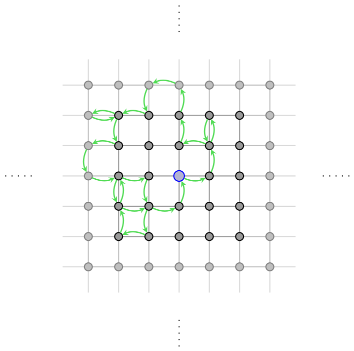
Infinite 3D Grid



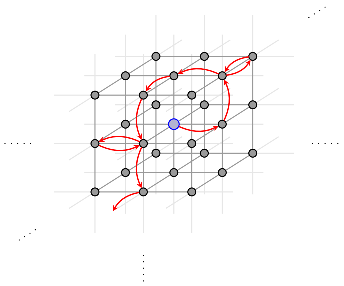
1921: The Birth of Random Walks on (Infinite) Graphs (Polyá)

Will a random walk always return to the origin?

Infinite 2D Grid



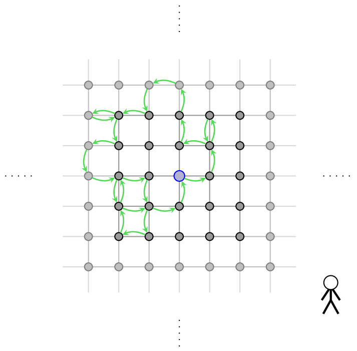
Infinite 3D Grid



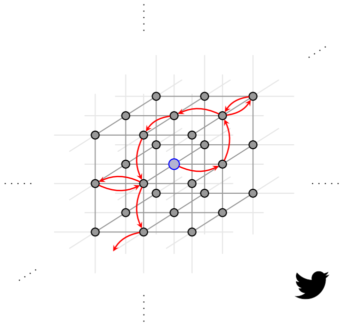
1921: The Birth of Random Walks on (Infinite) Graphs (Polyá)

Will a random walk always return to the origin?

Infinite 2D Grid



Infinite 3D Grid

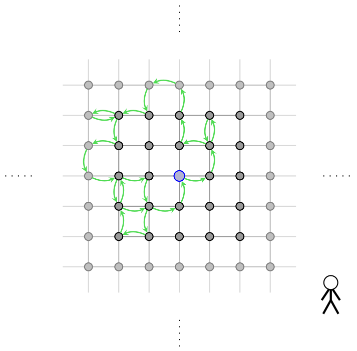


“A drunk man will find his way home, but a drunk bird may get lost forever.”

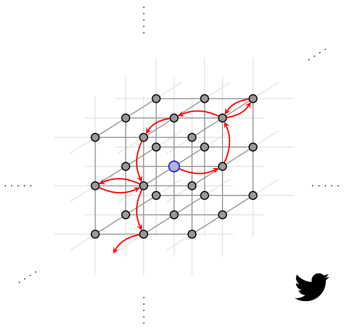
1921: The Birth of Random Walks on (Infinite) Graphs (Polyá)

Will a random walk always return to the origin?

Infinite 2D Grid



Infinite 3D Grid



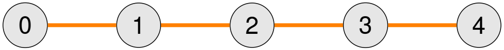
“A drunk man will find his way home, but a drunk bird may get lost forever.”

But for any regular (finite) graph, the expected return time to u is $1/\pi(u) = n$

SRW Random Walk on Two-Dimensional Grids: Animation

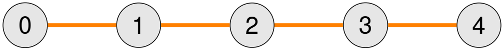
Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i, j\} : j = i + 1\}$.



Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i, j\} : j = i + 1\}$.

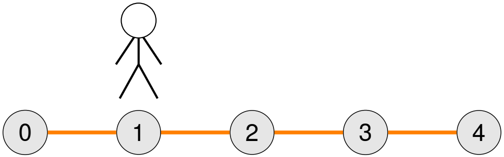


Proposition

For the SRW on P_n we have $h(k, n) = n^2 - k^2$, for any $0 \leq k < n$.

Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i, j\} : j = i + 1\}$.

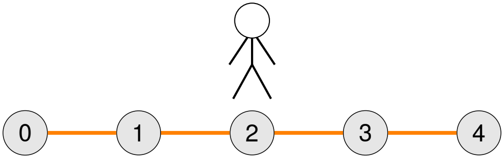


Proposition

For the SRW on P_n we have $h(k, n) = n^2 - k^2$, for any $0 \leq k < n$.

Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i, j\} : j = i + 1\}$.

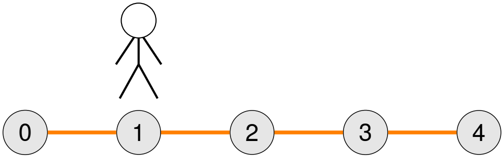


Proposition

For the SRW on P_n we have $h(k, n) = n^2 - k^2$, for any $0 \leq k < n$.

Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i,j\} : j = i+1\}$.

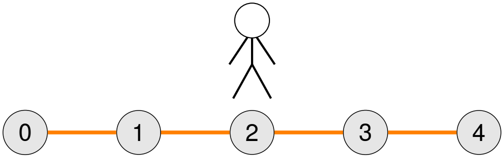


Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k < n$.

Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i, j\} : j = i + 1\}$.

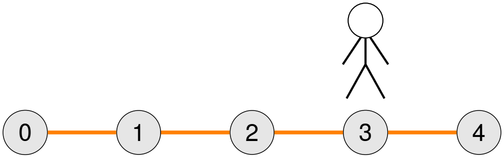


Proposition

For the SRW on P_n we have $h(k, n) = n^2 - k^2$, for any $0 \leq k < n$.

Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i, j\} : j = i + 1\}$.

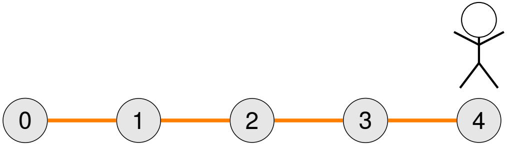


Proposition

For the SRW on P_n we have $h(k, n) = n^2 - k^2$, for any $0 \leq k < n$.

Random Walk on a Path (1/2)

The n-path P_n is the graph with $V(P_n) = [n]$ and $E(P_n) = \{\{i, j\} : j = i + 1\}$.



Proposition

For the SRW on P_n we have $h(k, n) = n^2 - k^2$, for any $0 \leq k < n$.

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Proof: Let $f(k) = h(k,n)$ and set $f(n) := 0$.

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Proof: Let $f(k) = h(k,n)$ and set $f(n) := 0$. By the Markov property

$f(0) = 1 + f(1)$

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Proof: Let $f(k) = h(k,n)$ and set $f(n) := 0$. By the Markov property

$$f(0) = 1 + f(1) \quad \text{and} \quad f(k) = 1 + \frac{f(k-1)}{2} + \frac{f(k+1)}{2} \quad \text{for } 1 \leq k \leq n-1.$$

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Proof: Let $f(k) = h(k,n)$ and set $f(n) := 0$. By the Markov property

$$f(0) = 1 + f(1) \quad \text{and} \quad f(k) = 1 + \frac{f(k-1)}{2} + \frac{f(k+1)}{2} \quad \text{for } 1 \leq k \leq n-1.$$

System of n independent equations in n unknowns, so has a unique solution.

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Proof: Let $f(k) = h(k,n)$ and set $f(n) := 0$. By the Markov property

$$f(0) = 1 + f(1) \quad \text{and} \quad f(k) = 1 + \frac{f(k-1)}{2} + \frac{f(k+1)}{2} \quad \text{for } 1 \leq k \leq n-1.$$

System of n independent equations in n unknowns, so has a unique solution.

Thus it suffices to check that $f(k) = n^2 - k^2$ satisfies the above.

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Proof: Let $f(k) = h(k,n)$ and set $f(n) := 0$. By the Markov property

$$f(0) = 1 + f(1) \quad \text{and} \quad f(k) = 1 + \frac{f(k-1)}{2} + \frac{f(k+1)}{2} \quad \text{for } 1 \leq k \leq n-1.$$

System of n independent equations in n unknowns, so has a unique solution.

Thus it suffices to check that $f(k) = n^2 - k^2$ satisfies the above. Indeed

$$f(0) = 1 + f(1) = 1 + n^2 - 1^2 = n^2,$$

Random Walk on a Path (2/2)

Proposition

For the SRW on P_n we have $h(k,n) = n^2 - k^2$, for any $0 \leq k \leq n$.

Recall: Hitting times are the solution to the set of linear equations:

$$h(x,y) \overset{\text{Markov Prop.}}{=} 1 + \sum_{z \in \Omega \setminus \{y\}} h(z,y) \cdot P(x,z) \qquad \forall x \neq y \in V.$$

Proof: Let $f(k) = h(k,n)$ and set $f(n) := 0$. By the Markov property

$$f(0) = 1 + f(1) \quad \text{and} \quad f(k) = 1 + \frac{f(k-1)}{2} + \frac{f(k+1)}{2} \quad \text{for } 1 \leq k \leq n-1.$$

System of n independent equations in n unknowns, so has a unique solution.

Thus it suffices to check that $f(k) = n^2 - k^2$ satisfies the above. Indeed

$$f(0) = 1 + f(1) = 1 + n^2 - 1^2 = n^2,$$

and for any $1 \leq k \leq n-1$ we have,

$$f(k) = 1 + \frac{n^2 - (k-1)^2}{2} + \frac{n^2 - (k+1)^2}{2} = n^2 - k^2. \qquad \square$$

Outline

SAT Problems

A Satisfiability (SAT) formula is a logical expression that's the conjunction (AND) of a set of Clauses, where a clause is the disjunction (OR) of Literals.

SAT Problems

A Satisfiability (SAT) formula is a logical expression that’s the conjunction (AND) of a set of Clauses, where a clause is the disjunction (OR) of Literals.

A Solution to a SAT formula is an assignment of the variables to the values `True` and `False` so that all the clauses are satisfied.

SAT Problems

A Satisfiability (SAT) formula is a logical expression that's the conjunction (AND) of a set of Clauses, where a clause is the disjunction (OR) of Literals.

A Solution to a SAT formula is an assignment of the variables to the values `True` and `False` so that all the clauses are satisfied.

Example:

SAT: $(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$

SAT Problems

A Satisfiability (SAT) formula is a logical expression that's the conjunction (AND) of a set of Clauses, where a clause is the disjunction (OR) of Literals.

A Solution to a SAT formula is an assignment of the variables to the values `True` and `False` so that all the clauses are satisfied.

Example:

SAT: $(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$

Solution: $x_1 = \texttt{True}$, $x_2 = \texttt{False}$, $x_3 = \texttt{False}$ and $x_4 = \texttt{True}$.

SAT Problems

A Satisfiability (SAT) formula is a logical expression that's the conjunction (AND) of a set of Clauses, where a clause is the disjunction (OR) of Literals.

A Solution to a SAT formula is an assignment of the variables to the values `True` and `False` so that all the clauses are satisfied.

Example:

$$\text{SAT: } (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

$$\text{Solution: } x_1 = \texttt{True}, \quad x_2 = \texttt{False}, \quad x_3 = \texttt{False} \quad \text{and} \quad x_4 = \texttt{True}.$$

- If each clause has k literals we call the problem k-SAT.
- In general, determining if a SAT formula has a solution is NP-hard
- In practice solvers are fast and used to great effect
- A huge amount of problems can be posed as a SAT:

SAT Problems

A Satisfiability (SAT) formula is a logical expression that's the conjunction (AND) of a set of Clauses, where a clause is the disjunction (OR) of Literals.

A Solution to a SAT formula is an assignment of the variables to the values `True` and `False` so that all the clauses are satisfied.

Example:

$$\text{SAT: } (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

$$\text{Solution: } x_1 = \texttt{True}, \quad x_2 = \texttt{False}, \quad x_3 = \texttt{False} \quad \text{and} \quad x_4 = \texttt{True}.$$

- If each clause has k literals we call the problem k-SAT.
- In general, determining if a SAT formula has a solution is NP-hard
- In practice solvers are fast and used to great effect
- A huge amount of problems can be posed as a SAT:
  - $\rightarrow$ Model checking and hardware/software verification
  - $\rightarrow$ Design of experiments
  - $\rightarrow$ Classical planning
  - $\rightarrow$ ...

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
```

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: **Repeat up to $2n^2$ times**
```

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
```

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
```

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return “Satisfiable”
6: return “Unsatisfiable”
```

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return “Satisfiable”
6: return “Unsatisfiable”
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.

2-SAT

```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** “Satisfiable”
6: **return** “Unsatisfiable”
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

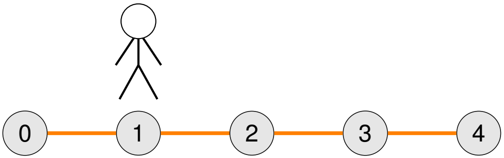
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F F T F T

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F

2-SAT

```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

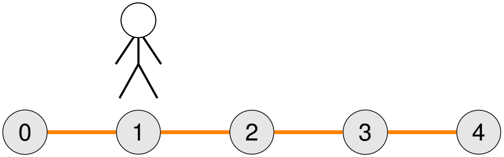
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F F T F T

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F

2-SAT

```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return “Satisfiable”
6: return “Unsatisfiable”
```

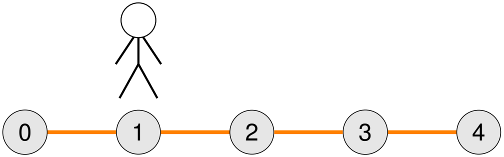
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F F T F T

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F

2-SAT

```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

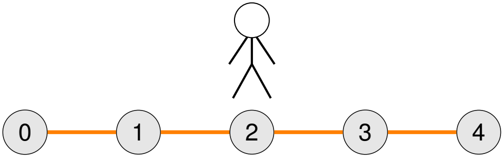
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

F F T T F T F T F T

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** “Satisfiable”
6: **return** “Unsatisfiable”
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

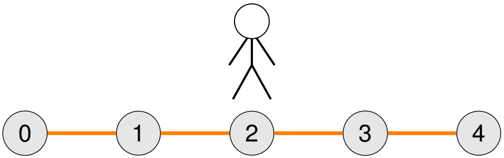
Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

F F T T F T F T F T

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$

t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F



2-SAT

```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** “Satisfiable”
6: **return** “Unsatisfiable”
```

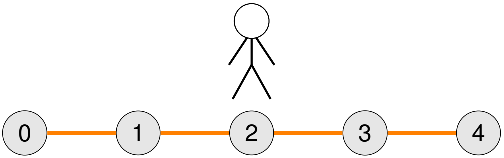
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

F F T T F T F T F T

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F

2-SAT

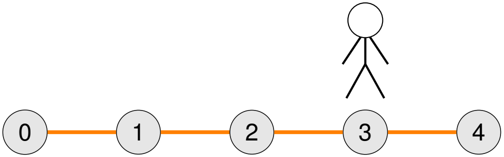
```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 1 :

$$\begin{array}{ccccccccccccccc}(x_1 & \vee & \overline{x_2}) & \wedge & (\overline{x_1} & \vee & \overline{x_3}) & \wedge & (x_1 & \vee & x_2) & \wedge & (x_4 & \vee & \overline{x_3}) & \wedge & (x_4 & \vee & \overline{x_1}) \\ \mathtt{T} & & \mathtt{F} & & \mathtt{F} & & \mathtt{T} & & \mathtt{T} & & \mathtt{T} & & \mathtt{F} & & \mathtt{T} & & \mathtt{F} & & \mathtt{F}\end{array}$$

$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F
2	T	T	F	F

2-SAT

```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** “Satisfiable”
6: **return** “Unsatisfiable”
```

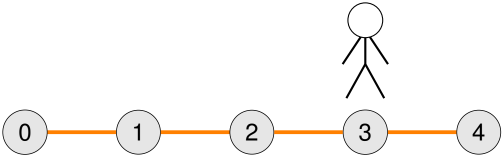
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

T F F T T T F T F F

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F
2	T	T	F	F

2-SAT

```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return “Satisfiable”
6: return “Unsatisfiable”
```

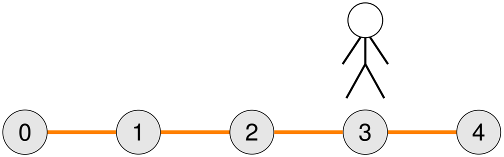
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

T F F T T T F T F F

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F
2	T	T	F	F

2-SAT

RANDOMISED2-SAT (Input: a 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

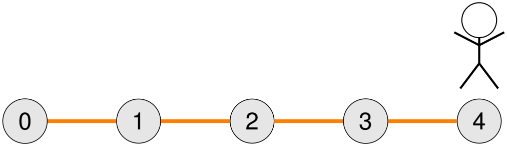
Example 1 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1})$$

T F F T T T T T T F

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$

t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F
2	T	T	F	F
3	T	T	F	T



2-SAT

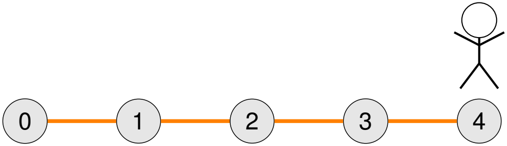
```
RANDOMISED2-SAT (Input: a 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clause
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 1 : Solution Found

$$\underset{\mathtt{T}}{(x_1} \vee \underset{\mathtt{F}}{\overline{x_2})} \wedge \underset{\mathtt{F}}{(\overline{x_1}} \vee \underset{\mathtt{T}}{\overline{x_3})} \wedge \underset{\mathtt{T}}{(x_1} \vee \underset{\mathtt{T}}{x_2)} \wedge \underset{\mathtt{T}}{(x_4} \vee \underset{\mathtt{T}}{\overline{x_3})} \wedge \underset{\mathtt{T}}{(x_4} \vee \underset{\mathtt{F}}{\overline{x_1})}$$

$$\alpha = (\mathtt{T}, \mathtt{T}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	T	F	F
2	T	T	F	F
3	T	T	F	T

2-SAT

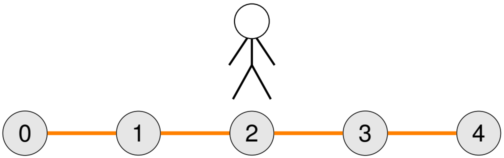
```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return “Satisfiable”
6: return “Unsatisfiable”
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 2 :

$$\begin{array}{ccccccccccccccc} (x_1 & \vee & \overline{x_2}) & \wedge & (\overline{x_1} & \vee & \overline{x_3}) & \wedge & (x_1 & \vee & x_2) & \wedge & (x_4 & \vee & x_3) & \wedge & (x_4 & \vee & \overline{x_1}) \\ \mathtt{F} & & \mathtt{T} & & \mathtt{T} & & \mathtt{T} & & \mathtt{F} & & \mathtt{F} & & \mathtt{F} & & \mathtt{F} & & \mathtt{F} & & \mathtt{T} \end{array}$$

$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T})$.



t	x_1	x_2	x_3	x_4
0	F	F	F	F

2-SAT

RANDOMISED2-SAT (Input: A 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** “Satisfiable”
6: **return** “Unsatisfiable”
```

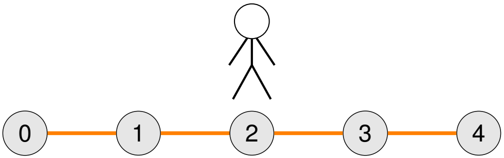
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F F F F T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F

2-SAT

```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

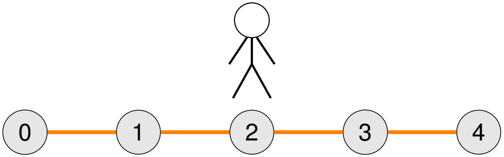
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F F F F T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F

2-SAT

```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

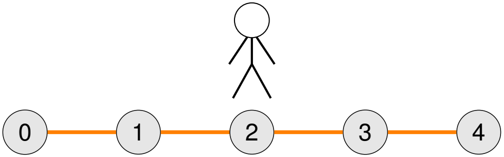
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F F F F T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F

2-SAT

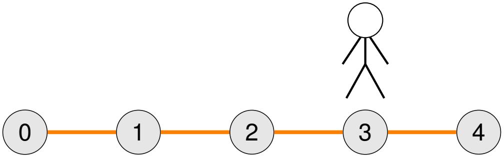
```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return “Satisfiable”
6: return “Unsatisfiable”
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 2 :

$$\underset{\mathtt{F}}{(x_1} \vee \underset{\mathtt{T}}{\overline{x_2})} \wedge \underset{\mathtt{T}}{(\overline{x_1}} \vee \underset{\mathtt{T}}{\overline{x_3})} \wedge \underset{\mathtt{F}}{(x_1} \vee \underset{\mathtt{F}}{x_2)} \wedge \underset{\mathtt{T}}{(x_4} \vee \underset{\mathtt{F}}{x_3)} \wedge \underset{\mathtt{T}}{(x_4} \vee \underset{\mathtt{T}}{\overline{x_1})}$$

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T

2-SAT

```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

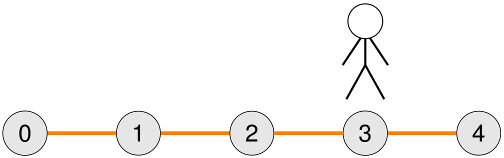
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F T F T T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T

2-SAT

```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** "Satisfiable"
6: **return** "Unsatisfiable"
```

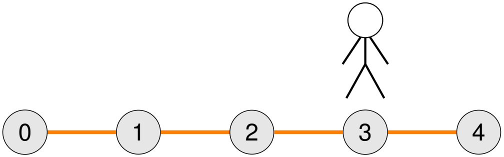
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F T T T F F T F T T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T

2-SAT

```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return “Satisfiable”
6: return “Unsatisfiable”
```

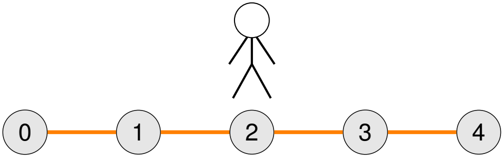
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F F T T F T T F T T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T
2	F	T	F	T

2-SAT

```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** "Satisfiable"
6: **return** "Unsatisfiable"
```

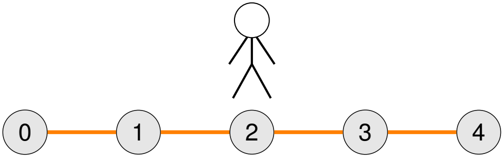
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F F T T F T T F T T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T
2	F	T	F	T

2-SAT

```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

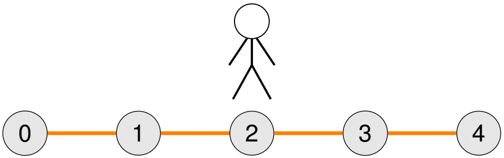
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

F F T T F T T F T T

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T
2	F	T	F	T

2-SAT

RANDOMISED2-SAT (Input: A 2-SAT-Formula)

```
1: Start with an arbitrary truth assignment
2: **Repeat up to** $2n^2$ **times**
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     **If** formula is satisfied **then return** “Satisfiable”
6: **return** “Unsatisfiable”
```

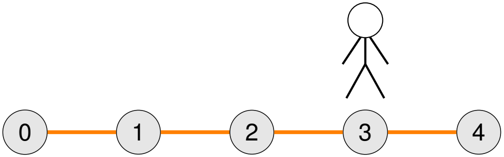
- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |$variable values shared by A_i and $\alpha|$.

Example 2 :

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee x_3) \wedge (x_4 \vee \overline{x_1})$$

T F F T T T T F T F

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T
2	F	T	F	T
3	T	T	F	T

2-SAT

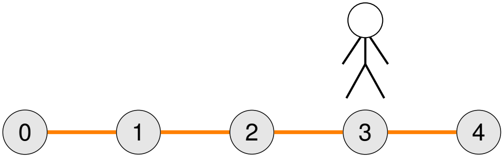
```
RANDOMISED2-SAT (Input: A 2-SAT-Formula)
1: Start with an arbitrary truth assignment
2: Repeat up to $2n^2$ times
3:     Pick an arbitrary unsatisfied clauses
4:     Choose a random literal and switch its value
5:     If formula is satisfied then return "Satisfiable"
6: return "Unsatisfiable"
```

- Call each loop of (2) a step. Let A_i be the variable assignment at step i.
- Let α be any solution and $X_i = |\text{variable values shared by } A_i \text{ and } \alpha|$.

Example 2 : (Another) Solution Found

$$\underset{\mathtt{T}}{(x_1} \vee \underset{\mathtt{F}}{\overline{x_2})} \wedge \underset{\mathtt{F}}{(\overline{x_1}} \vee \underset{\mathtt{T}}{\overline{x_3})} \wedge \underset{\mathtt{T}}{(x_1} \vee \underset{\mathtt{T}}{x_2)} \wedge \underset{\mathtt{T}}{(x_4} \vee \underset{\mathtt{F}}{x_3)} \wedge \underset{\mathtt{T}}{(x_4} \vee \underset{\mathtt{F}}{\overline{x_1})}$$

$$\alpha = (\mathtt{T}, \mathtt{F}, \mathtt{F}, \mathtt{T}).$$



t	x_1	x_2	x_3	x_4
0	F	F	F	F
1	F	F	F	T
2	F	T	F	T
3	T	T	F	T

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[X_{i+1} = 1 \mid X_i = 0] = 1$

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[\, X_{i+1} = 1 \mid X_i = 0 \,] = 1$

(ii) $\mathbf{P}[\, X_{i+1} = k+1 \mid X_i = k \,] \geq 1/2$

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[\, X_{i+1} = 1 \mid X_i = 0 \,] = 1$

(ii) $\mathbf{P}[\, X_{i+1} = k+1 \mid X_i = k \,] \geq 1/2$

(iii) $\mathbf{P}[\, X_{i+1} = k-1 \mid X_i = k \,] \leq 1/2$.

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[X_{i+1} = 1 \mid X_i = 0] = 1$

(ii) $\mathbf{P}[X_{i+1} = k+1 \mid X_i = k] \geq 1/2$

(iii) $\mathbf{P}[X_{i+1} = k-1 \mid X_i = k] \leq 1/2$.

Notice that if $X_i = n$ then $A_i = \alpha$ thus solution found (may find another first).

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[X_{i+1} = 1 \mid X_i = 0] = 1$

(ii) $\mathbf{P}[X_{i+1} = k+1 \mid X_i = k] \geq 1/2$

(iii) $\mathbf{P}[X_{i+1} = k-1 \mid X_i = k] \leq 1/2$.

Notice that if $X_i = n$ then $A_i = \alpha$ thus solution found (may find another first).

Assume (pessimistically) that $X_0 = 0$ (none of our initial guesses is right).

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[\, X_{i+1} = 1 \mid X_i = 0\,] = 1$

(ii) $\mathbf{P}[\, X_{i+1} = k+1 \mid X_i = k\,] \geq 1/2$

(iii) $\mathbf{P}[\, X_{i+1} = k-1 \mid X_i = k\,] \leq 1/2$.

Notice that if $X_i = n$ then $A_i = \alpha$ thus solution found (may find another first).

Assume (pessimistically) that $X_0 = 0$ (none of our initial guesses is right).

The stochastic process X_i is complicated to describe in full; however by $(i) - (iii)$ we can **bound** it by Y_i (SRW on the n-path from 0).

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[X_{i+1} = 1 \mid X_i = 0] = 1$

(ii) $\mathbf{P}[X_{i+1} = k+1 \mid X_i = k] \geq 1/2$

(iii) $\mathbf{P}[X_{i+1} = k-1 \mid X_i = k] \leq 1/2$.

Notice that if $X_i = n$ then $A_i = \alpha$ thus solution found (may find another first).

Assume (pessimistically) that $X_0 = 0$ (none of our initial guesses is right).

The stochastic process X_i is complicated to describe in full; however by $(i) - (iii)$ we can **bound** it by Y_i (SRW on the n-path from 0). This gives

$$\mathbf{E}[\text{time to find sol}] \leq \mathbf{E}_0[\min\{t : X_t = n\}] \leq \mathbf{E}_0[\min\{t : Y_t = n\}] = h(0,n) = n^2.$$

□

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[X_{i+1} = 1 \mid X_i = 0] = 1$

(ii) $\mathbf{P}[X_{i+1} = k+1 \mid X_i = k] \geq 1/2$

(iii) $\mathbf{P}[X_{i+1} = k-1 \mid X_i = k] \leq 1/2$.

Notice that if $X_i = n$ then $A_i = \alpha$ thus solution found (may find another first).

Assume (pessimistically) that $X_0 = 0$ (none of our initial guesses is right).

The stochastic process X_i is complicated to describe in full; however by $(i) - (iii)$ we can **bound** it by Y_i (SRW on the n-path from 0). This gives

$$\mathbf{E}[\text{time to find sol}] \leq \mathbf{E}_0[\min\{t : X_t = n\}] \leq \mathbf{E}_0[\min\{t : Y_t = n\}] = h(0,n) = n^2.$$

$\square$

Running for $2n^2$ time and using Markov's inequality yields:

2-SAT and the SRW on the Path

Expected iterations of (2) in RANDOMISED2-SAT

If the formula is satisfiable, then the expected number of steps before RANDOMISED2-SAT outputs a valid solution is at most n^2.

Proof: Fix any solution α, then for any $i \geq 0$ and $1 \leq k \leq n-1$,

(i) $\mathbf{P}[X_{i+1} = 1 \mid X_i = 0] = 1$

(ii) $\mathbf{P}[X_{i+1} = k+1 \mid X_i = k] \geq 1/2$

(iii) $\mathbf{P}[X_{i+1} = k-1 \mid X_i = k] \leq 1/2$.

Notice that if $X_i = n$ then $A_i = \alpha$ thus solution found (may find another first).

Assume (pessimistically) that $X_0 = 0$ (none of our initial guesses is right).

The stochastic process X_i is complicated to describe in full; however by $(i) - (iii)$ we can **bound** it by Y_i (SRW on the n-path from 0). This gives

$$\mathbf{E}[\text{time to find sol}] \leq \mathbf{E}_0[\min\{t : X_t = n\}] \leq \mathbf{E}_0[\min\{t : Y_t = n\}] = h(0,n) = n^2.$$

$\square$

Running for $2n^2$ time and using Markov's inequality yields:

Proposition

Provided a solution exists, RANDOMISED2-SAT will return a valid solution in $O(n^2)$ time with probability at least $1/2$.

Boosting Success Probabilities

Boosting Lemma

Suppose a randomised algorithm succeeds with probability (at least) p. Then for any $C \geq 1$, $\lceil \frac{C}{p} \cdot \log n \rceil$ repetitions are sufficient to succeed (in at least one repetition) with probability at least $1 - n^{-C}$.

Boosting Success Probabilities

Boosting Lemma

Suppose a randomised algorithm succeeds with probability (at least) p. Then for any $C \geq 1$, $\lceil \frac{C}{p} \cdot \log n \rceil$ repetitions are sufficient to succeed (in at least one repetition) with probability at least $1 - n^{-C}$.

Proof: Recall that $1 - p \leq e^{-p}$ for all real p. Let $t = \lceil \frac{C}{p} \log n \rceil$ and observe

$$\begin{aligned} \mathbf{P}[\, t \text{ runs all fail} \,] &\leq (1-p)^t \\ &\leq e^{-pt} \\ &\leq n^{-C}, \end{aligned}$$

thus the probability one of the runs succeeds is at least $1 - n^{-C}$. $\square$

Boosting Success Probabilities

Boosting Lemma

Suppose a randomised algorithm succeeds with probability (at least) p. Then for any $C \geq 1$, $\lceil \frac{C}{p} \cdot \log n \rceil$ repetitions are sufficient to succeed (in at least one repetition) with probability at least $1 - n^{-C}$.

Proof: Recall that $1 - p \leq e^{-p}$ for all real p. Let $t = \lceil \frac{C}{p} \log n \rceil$ and observe

$$\begin{aligned} \mathbf{P}[\, t \text{ runs all fail} \,] &\leq (1-p)^t \\ &\leq e^{-pt} \\ &\leq n^{-C}, \end{aligned}$$

thus the probability one of the runs succeeds is at least $1 - n^{-C}$. □

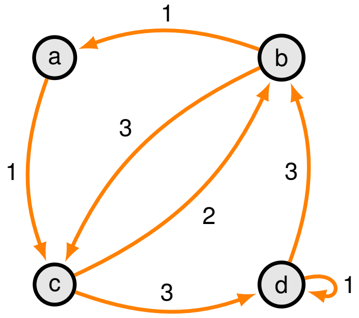
RANDOMISED2-SAT

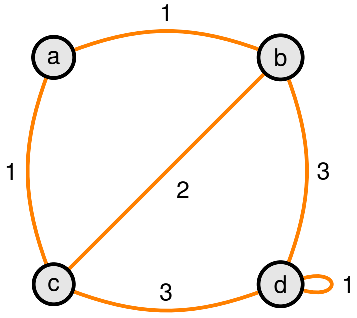
There is a $O(n^2 \log n)$-time algorithm for 2-SAT which succeeds w.h.p.

Outline

Random Walks on Weighted Graphs

An (edge) weighted graph $G = (V, E, w)$ where $w : E \to \mathbb{R}_+$ on the edges.



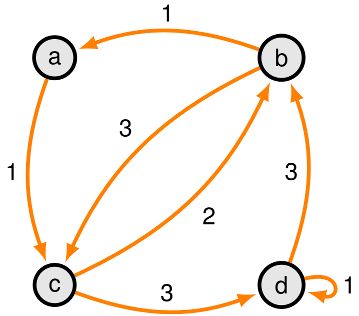


Random Walks on Weighted Graphs

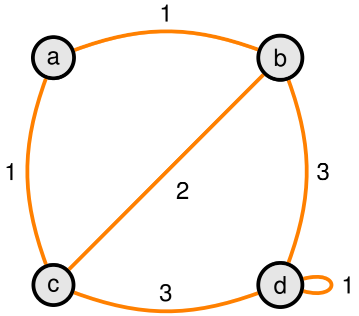
An (edge) weighted graph $G = (V, E, w)$ where $w : E \to \mathbb{R}_+$ on the edges.

A Simple Random Walk (SRW) on a weighted graph G is a MC on $V(G)$ with

$$P(i,j) = \begin{cases} \frac{w(i,j)}{\sum_{\{x,y\} \in E} w(x,y)} & \text{if } \{i,j\} \in E \\ 0 & \text{if } \{i,j\} \notin E \end{cases}.$$



Directed

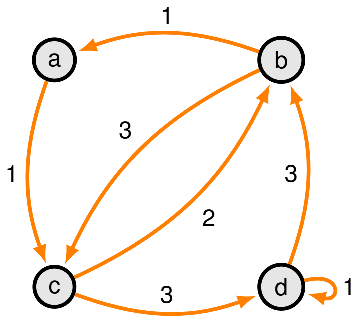


Random Walks on Weighted Graphs

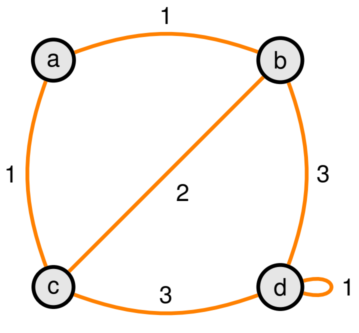
An (edge) weighted graph $G = (V, E, w)$ where $w : E \to \mathbb{R}_+$ on the edges.

A Simple Random Walk (SRW) on a weighted graph G is a MC on $V(G)$ with

$$P(i,j) = \begin{cases} \frac{w(i,j)}{\sum_{\{x,y\}\in E} w(x,y)} & \text{if } \{i,j\} \in E \\ 0 & \text{if } \{i,j\} \notin E \end{cases}.$$



Directed



Undirected

Reversible Markov Chains

- Any Markov Chain can be described as random walk on a weighted **directed** graph.

Reversible Markov Chains

- Any Markov Chain can be described as random walk on a weighted **directed** graph.

Definition

A Markov chain on Ω with transition matrix P and stationary distribution π is called reversible if, for any $x, y \in \Omega$,

$$\pi(x)P(x,y) = \pi(y)P(y,x)$$

Reversible Markov Chains

- Any Markov Chain can be described as random walk on a weighted **directed** graph.

Definition

A Markov chain on Ω with transition matrix P and stationary distribution π is called reversible if, for any $x, y \in \Omega$,

$$\pi(x)P(x,y) = \pi(y)P(y,x)$$

- Reversible Markov Chains are equivalent to random walks on weighted **undirected** graphs.

Reversible Markov Chains

- Any Markov Chain can be described as random walk on a weighted **directed** graph.

Definition

A Markov chain on Ω with transition matrix P and stationary distribution π is called reversible if, for any $x, y \in \Omega$,

$$\pi(x)P(x, y) = \pi(y)P(y, x)$$

- Reversible Markov Chains are equivalent to random walks on weighted **undirected** graphs.
- A reversible Markov Chain identified with the (undirected) weighted graph $G = (V, E, w)$ has stationary distribution given by

$$\pi(i) = \frac{\sum_{j:\{i,j\}\in E} w(i,j)}{2\sum_{\{x,y\}\in E} w(x,y)}$$