1 Answers

**task 0**) Take \(a_1 = 1, a_2 = -1, g_1 = n, g_2 = -n\). Then \(n \in O(a_1 g_1 + a_2 g_2) = O(2n)\) but \(n \notin O(g_1 + g_2) = O(0)\) (\(O(0)\) is the set of functions that are eventually 0).

**task 1**) \(O(n)\) is the set of functions eventually bounded by \(n\) to within a constant factor. So to show that \(n^2 \notin O(n)\) we must show that for any \(n_0 \in \mathbb{N}\) and any \(c \in \mathbb{R}_{>0}\) there exists \(n \geq n_0\) such that \(n^2 > cn\) (i.e. however far out we go on the x-axis (any \(n_0\)), and whatever constant we choose (any \(c\)), we can always find somewhere further along \((n \geq n_0)\) where \(n^2 > cn\). To prove this, fix \(n_0 \in \mathbb{N}\) and \(c \in \mathbb{R}_{>0}\) and let \(n := \max(n_0, \lfloor c \rfloor + 1)\). Then \(n > c\) so \(n^2 > cn\).

**task 2**) To prove this, fix \(n_0 \in \mathbb{N}\) and \(c \in \mathbb{R}_{>0}\) and let 
\[
i := \lceil c \rceil + 1 \quad n := \max(n_0 + 1, i! + 1)
\]

Then
\[
i! < i! + 1 \leq n \leq n^{i-c}
\]

Therefore, \(n^c < \frac{n^i}{i^c} < e^n\) (since \(\frac{n^i}{i^c}\) is a term in the Maclaurin expansion of \(e^n\)). Since \(\log\) is monotone increasing (i.e. preserves strict inequalities), we can take the \(\log\) of both sides and we have \(\log n < n\).

**task 3**) Suppose that \(f, g \in O(h)\) and \(a, b \in \mathbb{R}\). Then there exist \(n_0, m_0 \in \mathbb{N}\) and \(c_0, d_0 \in \mathbb{R}_{>0}\) such that for all \(n \geq \max(n_0, m_0)\), \(|f(n)| \leq c_0|h(n)|\) and \(|g(n)| \leq d_0|h(n)|\). Then, by the triangle inequality, for all \(n \geq \max(n_0, m_0)\)
\[
|\alpha f(n) + \beta g(n)| \leq |\alpha||f(n)| + |\beta||g(n)|
\]
\[
\leq (|\alpha|c_0 + |\beta|d_0)|h(n)|
\]
hence \(\alpha f + \beta g \in O(h)\). Since \(0 \in O(h)\), this shows that \(O(h)\) is closed under linear operations.

To show that, for \(a > 0\), \(an + b \log n + c \notin O(\log n)\) we assume that it is and then derive a contradiction. We know that \(1 \in O(\log n)\) and that \(\log n \in O(\log n)\) so \(b \log n + c \in O(\log n)\) (since \(O(\log n)\) is closed under linear operations). Hence (using that \(O(\log n)\) is closed under linear operations again)
\[
n = \frac{1}{a}((an + b \log n + c) - (b \log n + c)) \in O(\log n)
\]
which is false.

**task 4.5**) Let \(N(n, l)\) be the number of ways of making change for \(n\) from a set of coins \(l\). Let’s calculate \(N(n, \{a, b\})\) where \(a \neq b\). What exactly are we counting? A ‘way’ of making change is a function \(f : \{a, b\} \rightarrow \mathbb{N}\) such that \(af(a) + b \cdot f(b) = n\). Such a function is also known as a multiset\(^1\) (like a list where we don’t care about the order of its elements). So
\[
N(n, \{a, b\}) = |\{f : \{a, b\} \rightarrow \mathbb{N} : af(a) + bf(b) = n\}|
\]

\(^1\)http://en.wikipedia.org/wiki/Multiset#Formal_definition
where \(|X|\) is the size (cardinality) of the set \(X\). We have

\[
\{ f : \{a, b\} \to \mathbb{N} : af(a) + bf(b) = n \} = \\
\bigcup_{i=0}^{\lfloor n/a \rfloor} \{ f : \{a, b\} \to \mathbb{N} : f(a) = i \land bf(b) = n - ai \}
\]

(to prove that sets \(X\) and \(Y\) are equal, you show that \(x \in X \iff x \in Y\)). Now the sets on the right are mutually disjoint so

\[
\bigg| \bigcup_{i=0}^{\lfloor n/a \rfloor} \{ f : \{a, b\} \to \mathbb{N} : f(a) = i \land bf(b) = n - ai \} \bigg| = \\
\sum_{i=0}^{\lfloor n/a \rfloor} |\{ f : \{a, b\} \to \mathbb{N} : f(a) = i \land bf(b) = n - ai \}|
\]

and for \(0 \leq i \leq \lfloor n/a \rfloor\)

\[
|\{ f : \{a, b\} \to \mathbb{N} : f(a) = i \land bf(b) = n - ai \}| = |\{ f : \{b\} \to \mathbb{N} : bf(b) = n - ai \}|
\]

so

\[
N(n, \{a, b\}) = \sum_{i=0}^{\lfloor n/a \rfloor} N(n - ia, \{b\})
\]

and, from the formal definition of \(N\), we can show that for all \(m\)

\[
N(m, \{b\}) = \begin{cases} 
1 & \text{if } b|m \\
0 & \text{else}
\end{cases}
\]

hence

\[
N(n, \{a, b\}) \leq \sum_{i=0}^{\lfloor n/a \rfloor} 1 = \lfloor n/a \rfloor + 1 \leq n/a + 1 \in O(n)
\]

We can generalize this argument to show by induction that, for \(k \geq 2, \forall n \in \mathbb{N} \cdot N(n, \{a_1, \ldots, a_k\}) \leq (n+1)^{k-1} \). This is true for \(k = 1\). Suppose true for \(k \geq 1\). Then

\[
N(n, \{a_1, \ldots, a_{k+1}\}) = \sum_{i=0}^{\lfloor n/a_{k+1} \rfloor} N(n - a_{k+1}i, \{a_1, \ldots, a_k\}) \quad \text{by generalizing the}
\]

\[
\leq \sum_{i=0}^{\lfloor n/a_{k+1} \rfloor} (n - a_{k+1}i + 1)^k \quad \text{by induction hypothesis}
\]

so \(N(n, \{a_1, \ldots, a_{k+1}\}) \in O(n^k)\). In fact, if \(a\) and \(b\) are co-prime then \(N(n, \{a, b\}) \in \Omega(n)\) as well. By Bézout’s Identity, there exist positive integers \(x, y\) such that \(ax - by = 1\). Then for all \(k \in \mathbb{Z}\), \(a(nx - kb) + b(-ny + ka) = n\). This gives us solutions when

\[
\frac{ny}{a} \leq k \leq \frac{nx}{b}
\]

which gives us at least \(n(\frac{x}{b} - \frac{y}{a}) - 2 = \frac{n}{ab} - 2\) solutions.
2 Structural Induction

Let’s define size \( t \) to be the number of values in a tree \( t \). So

\[
\text{size \( Lf \) = 0}
\]
\[
\text{size \( \text{Br} (v,t_l,t_r) \) = 1 + (size \( t_l \)) + (size \( t_r \))}
\]

Now let’s show by structural induction that

\[
\text{size \( t \) = length \( \text{flatten} \( t \) \)}
\]

Since there are two constructors for the binary tree datatype, there are two cases to consider. First, \( Lf \)

\[
\text{size \( Lf \) = 0} = \text{length} \[ \] = \text{length} \( \text{flatten \( Lf \) \)}
\]

Now the case for constructor \( \text{Br} \). Assume the theorem is true for trees \( t_l \) and \( t_r \). Then show true for \( \text{Br} (v,t_l,t_r) \) for all values \( v \):

\[
\text{size} (\text{Br} (v,t_l,t_r)) = 1+(\text{size} \( t_l \)) + (\text{size} \( t_r \))
\]
\[
= 1+(\text{length} \( \text{flatten} \( t_l \) \)) + (\text{length} \( \text{flatten} \( t_r \) \))
\]
\[
= \text{length} ((\text{flatten} \( t_l \))@[v]@(\text{flatten} \( t_r \) ))
\]
\[
= \text{length} (\text{flatten} (\text{Br} (v,t_l,t_r)))
\]

3 Tasks

**task 1)** Prove by structural induction that for all lists \( vs \)

\[
\text{inord} (t,vs) = (\text{inorder} \( t \))@vs
\]

as claimed on p78 of the notes. Thanks to referential transparency, we may now replace \( \text{inord} (t,[]) \) for \( \text{inorder} \( t \) \) anywhere in our ML code.

**task 2)** Questions 7.1 and 7.2 from the notes

**task 3)** Prove by structural induction that, for all \( v \), \( T (t,v) = \text{size} (t) \) where \( \text{size} (t) \) is the number of values in the tree \( t \) and \( T (t,v) \) is the number of cons operations that have been done in the evaluation of \( \text{inord} (t,v) \). That is, prove that \( \text{inord} \) has time complexity \( O(n) \).

**task 4)** Let \( t \) be the tree of size \( n \) with all non-leaf sub-trees on the left. Show that \( T (t) = h(n) \) for some \( h \in \Omega(n^2) \) and \( T (t) \) is the number of cons operations that were done in the evaluation of \( \text{inorder} \( t \) \). That is, prove that \( \text{inorder} \) requires time \( O(n^2) \) in this special case.

**task 5)** Question 8.5 from the notes (and 8.6 if you feel like it).

**task 6)** Question 10.2 from the notes.