CSC 236 HW 3

Check in period: 6/22 - 6/25

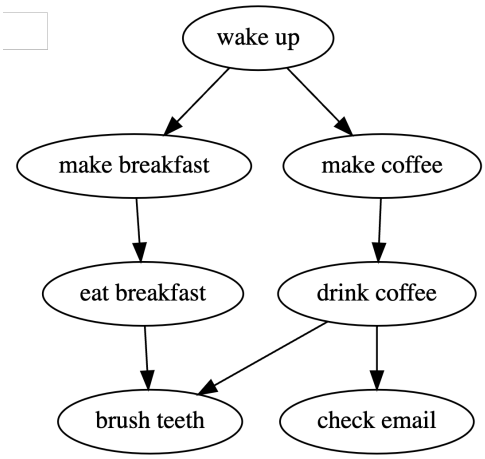
About

Please see the guide to hw, and guide to check ins for information and tips for the HW and the check ins!

These homeworks are on lectures and tutorials 1-6 inclusive.

1 Task Scheduling

Suppose you had a bunch of tasks. Here's a small example:



In the problem, we'll study the problem of finding a valid ordering of the tasks.

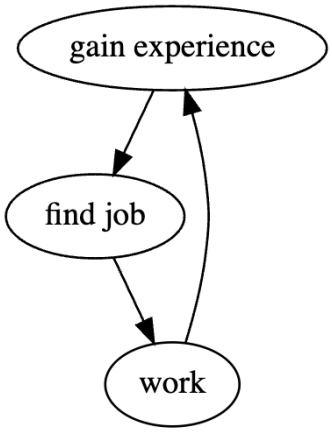
Call task a a prerequisite for task b if task a must be done before b, and write $a \prec b$.

Let T be a set of n tasks. Let $t_1, ..., t_n$ be an ordering of the elements in T. We say that the ordering respects $\prec$ or is valid if there does not exists $i, j \in [n]$ such that $i < j$ and $t_j \prec t_i$. In other words, if t_j appears after t_i, t_j is not a prerequisite for t_i.

Question 1a. Give two orderings that respect the morning schedule example (in that example $a \prec b$ if there is a directed edge from a to b).

- Wake up, make breakfast, make coffee, eat breakfast, drink coffee, brush teeth, check email
- Wake up, make coffee, drink coffee, check email, make breakfast, eat breakfast, brush teeth

Question 1b. Argue that there is no valid ordering for the following example



Since each of the tasks has a prerequisite, none of them can be the first task.

Let's impose some additional constraints what kind of relationships we can allow. In particular, if $\prec$ is a relation satisfying the following: $\forall a, b, c \in T$.

- $a \not\prec a$. That is, no task is a prerequisites of itself.
- If $a \prec b$, and $b \prec c$, then $a \prec c$. That is, a prerequisite of a prerequisite is also a prerequisite.

call $\prec$ a valid prerequisite relation over V.

Call a task $a \in T$ minimal if it does not have any prerequisites.

Question 1c. Show that $\forall n \geq 1$, if T is a set of n tasks and $\prec$ is a valid prerequisite relation over T, then T has a minimal element.

By induction on the number of elements.

Let $P(n)$ be the following predicate: For every set T on n elements and every valid prerequisites relation over T, T has a minimal element.

Base case. For $n = 1$, T contains a single element a, since no task is a prerequisite of itself, a has no prerequisites.

Inductive step. Let $k \in \mathbb{N}$ with $k \geq 1$, and assume $P(k)$. Now let T be a set on $k + 1$ elments and $\prec$ be a valid prerequisite relation over T. Let a be any element in T, and consider the set $T' = T \setminus \{a\}$. Note that T' is a set of k elements, and furthmore note that $\prec$ is still a valid prerequisite relation over T'. Thus, by the inductive hypothesis, T' has a minimal element b.

If $a \not\prec b$, then b is still minimal in T, and we're done. Otherwise, $a \prec b$. In this case, we claim that a is minimal in T. By contradiction, assume a was not minimal. Then there is

some $c \in T$ such that $c \prec a$. Since $c \prec a$, and $a \prec b$, we have $c \prec b$, which contradicts the minimality of b in T'.

Question 1d. Show that for any finite, non-empty set of tasks T, and valid prerequisite structure $\prec$ on T, there is a valid ordering of the tasks in T.

We'll show that $\forall n \in \mathbb{N}$ with $n \geq 1$, if T is a set of size n, and $\prec$ is a valid prerequisite relation over T, then there exists a valid ordering. By induction.

Base case. Let T be a set of size 1. T contains a single element a and this is already an ordering that respects $\prec$.

Inductive step. Let $k \in \mathbb{N}$ with $k \geq 1$, and assume the claim holds for sets of size k. Let T be a set of size $k+1$. By part 4c, T has a minimal element a. Let $T' = T \setminus \{a\}$. By the inductive hypothesis, T' has a ordering $(t_1, ..., t_k)$ that respects $\prec$. We claim that $(a, t_1, ..., t_k)$ is an ordering of elements of T that respects $\prec$. Since $(t_1, ..., t_k)$ was a valid ordering of T', we just need to check that a does not have a prerequisite among $t_1, ..., t_k$. This is guaranteed by the fact that a is minimal in T.

2 Propositional Formula Equivalence

Recall that a propositional formula is built up of propositioal variables (letters that represent arbitrary propositions, statements that are either True or False) using propositional connectives like negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), implication ($\implies$), etc. Also recall that two propositional formulae ϕ_1, ϕ_2 are said to be logically equivalent (denoted $\phi_1 \equiv \phi_2$) when they have the same value for every assignment of True/False to their variables. In case you need a refresher, you can find an overview of these notions here: https://en.wikipedia.org/wiki/Logical_equivalence.

Let B be the set of propositional variables. Define the following functions $\mathbf{E}_\neg, \mathbf{E}_\vee, \mathbf{E}_\wedge$ as follows,

- $\mathbf{E}_\neg(f) = \neg f$
- $\mathbf{E}_\vee(f, g) = f \vee g$
- $\mathbf{E}_\wedge(f, g) = f \wedge g$

Let G be the set generated from B by the functions $\{\mathbf{E}_\neg, \mathbf{E}_\vee, \mathbf{E}_\wedge\}$.

Question 2a. Prove that for every $f \in G$, there exists $f' \in G$ such that f and f' are logically equivalent, and f' does not contain the $\wedge$ symbol.

By structural induction.

Base case. Let $f = x$, where x is a propositional variable. Then x is equivalent to itself, and does not contain the $\wedge$ symbol.

Inductive step. Assume $f_1, f_2 \in G$, and suppose $P(f_1)$ and $P(f_2)$, i.e., there exist $f_1', f_2' \in G$ such that f_1' is logically equivalent to f_1, and f_2' is logically equivalent to f_2, and f_1' and f_2' do not contain the $\wedge$ symbol.

We'll show $P(\neg f_1)$, $P((f_1 \vee f_2))$, and $P((f_1 \wedge f_2))$.

- $\neg f_1'$ is a propositional formula equivalent to $\neg f_1$ that does not contain $\wedge$.
- $f_1' \vee f_2'$ is a propositional formula equivalent to $f_1 \vee f_2$ that does not contain $\wedge$.
- We claim that $\neg(\neg f_1' \vee \neg f_2')$ is logically equivalent to $f_1 \wedge f_2$. Indeed,

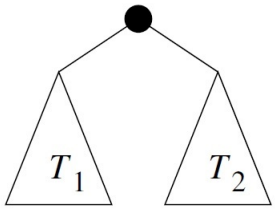
$$\begin{aligned} \neg(\neg f_1' \vee \neg f_2') &\equiv \neg\neg f_1' \wedge \neg\neg f_2' && \text{(De Morgan's Law)} \\ &\equiv f_1' \wedge f_2' && (\neg\neg f \equiv f) \\ &\equiv f_1 \wedge f_2 && (IH). \end{aligned}$$

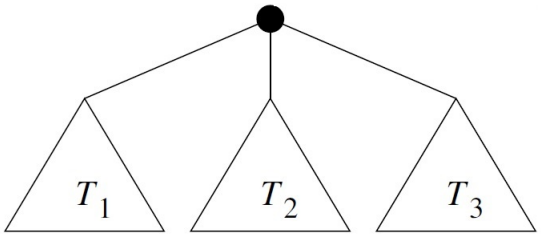
  Furthermore, since f_1' and f_2' don't contain $\wedge$, neither does $\neg(\neg f_1' \vee \neg f_2')$. This completes the induction.

3 Ternary Trees

Consider a subset $\mathbb{B}$ of binary trees recursively defined as follows:

- a single node is a member of $\mathbb{B}$;
- if $T_1, T_2 \in \mathbb{B}$ and T_1, T_2 have *the same height*, then the tree constructed by placing T_1 and T_2 under a new root node (as illustrated below on the left) is also a member of $\mathbb{B}$;
- nothing else belongs to $\mathbb{B}$.





Also define a subset $\mathbb{C}$ of ternary trees as follows:

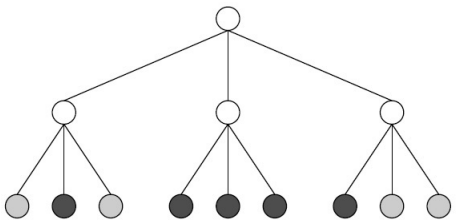
- a single node is a member of $\mathbb{C}$;
- if $T_1, T_2, T_3 \in \mathbb{C}$ and T_1, T_2, T_3 have *the same height*, then the tree constructed by placing T_1, T_2, and T_3 under a new root node (as illustrated above on the right) is also a member of $\mathbb{C}$;
- nothing else belongs to $\mathbb{C}$.

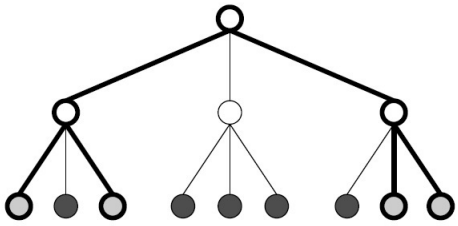
Finally, we define the following terms.

- A coloured tree is a tree in which every leaf node has been assigned one of two colours (red or blue). Only the leaves are coloured - internal nodes are colourless.
- A leaves-subtree of a tree T is a subset of T's leaves along with all of the leaves' ancestors and the edges connecting them. Note that every ternary tree contains many binary leaves-subtrees.
- A monochromatic binary tree is a coloured binary tree all of whose leaves are the same colour.

Question 3a.

Prove that every colored tree $T \in \mathbb{C}$ contains at least one monochromatic leaves-subtree which is a member of $\mathbb{B}$. (For example, the coloured ternary tree on the left below, which is a member of $\mathbb{C}$, contains a leaves-subtree, indicated on the right by thick edges and node outlines, which is monochromatic and is a member of $\mathbb{B}$.)





Note that you cannot choose how the ternary tree is coloured. Rather, your proof must establish that the above statement is valid no matter how colours are assigned to the leaves of the ternary tree.

$P(T)$: arbitrarily coloured tree T contains a leaves-subtree $T' \in \mathbb{B}$ which is monochromatic.

The goal is to prove that for all $T \in \mathbb{C}$, $P(T)$ holds. Proof by structural induction on $\mathbb{C}$.

Base Case: Let T be a single node. A single node is a member of $\mathbb{B}$, and has a single colour assigned to it. Therefore, T is a member of $\mathbb{B}$ which is monochromatic. Moreover, it is a leaves-subtree of itself. Therefore, T contains a leaves-subtree which is a member of $\mathbb{B}$ and is monochromatic, and so $P(T)$ holds.

Inductive Step: Let $T_1, T_2, T_3 \in \mathbb{C}$, and have the same height. By definition, $\mathbb{C}$ includes a ternary tree T consisting of a root with T_1, T_2 and T_3 as subtrees.
Suppose $P(T_1), P(T_2)$, and $P(T_3)$.

We'll show $P(T)$ holds. Since T_1, T_2 and T_3 have the same height, they contain binary leaves-subtrees of the same height. By IH, each of T_1, T_2 and T_3 respectively contain leaves-subtrees T_1', T_2', T_3' which are monochromatic and are members of $\mathbb{B}$. That is, $T_1', T_2', T_3' \in \mathbb{B}$, T_1', T_2', T_3' have the same height, and all leaves of each T_1', T_2' and T_3' are coloured either red or blue.
Since there are 2 colours and 3 trees, by the pigeonhole principle, at least two of these monochromatic leaves-subtrees have to be of the same colour. Without loss of generality, let's assume that all leaves of both T_1' and T_2' are red. In this case, T has a binary leaves-subtree T' that is red. Namely, the one obtained by placing T_1' and T_2' under the root of T. Since the root of T is not contained in T_1 or T_2, T' is a member of $\mathbb{B}$, by the recursive definition of $\mathbb{B}$. By construction, T' is a leaves-subtree of T and monochromatically red. Thus, $P(T)$. So, by structural induction, for all $T \in \mathbb{C}$, $P(T)$ holds.

4 Approximating Square Root of 2

It is useful to approximate irrational numbers using rational numbers. For example, a standard approximation for π is $22/7$. For more on this interesting topic, check out this video from Numberphile (optional). In this problem, we'll use a recurrence to find some approximations for $\sqrt{2}$.

Let $P(n)$ be the following recurrence.

$$P(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ 2P(n-1) + P(n-2) & n > 1 \end{cases}$$

Note. It is important here that you do the problems in order. For example, you may not use the result of the third subpart in your solution to the second subpart.

Question 4a. What are $P(0), P(1), ..., P(9)$?

0, 1, 2, 5, 12, 29, 70, 169, 408, 985

Question 4b. Find r such that $P(n) = \Theta(r^n)$, and prove that your choice in r is correct.

Hint: Set up a quadratic equation as we did in lecture 6.

$r = (1 + \sqrt{2}) \approx 2.414$. Note that r is the positive root of the quadratic $x^2 - 2x - 1$. Thus $r^2 = 2r + 1$.

First, we'll show $P(n) = O(r^n)$. Use the substitution method.

$$\begin{aligned} P(n) &= 2P(n-1) + P(n-2) \\ &\leq 2cr^{n-1} + cr^{n-2} \\ &= (cr^{n-2})(2r+1) \\ &= (cr^{n-2})r^2 \\ &= cr^n \end{aligned}$$

Next, we'll show $P(n) = \Omega(r^n)$. Use the substitution method.

$$\begin{aligned} P(n) &= 2P(n-1) + P(n-2) \\ &\geq 2cr^{n-1} + cr^{n-2} \\ &= (cr^{n-2})(2r+1) \\ &= (cr^{n-2})r^2 \\ &= cr^n \end{aligned}$$

Setting up the quadratic. The inductive step for a proof $P(n) = O(x^n)$ looks something like

$$\begin{aligned} P(n) &= 2P(n-1) + P(n-2) \\ &\leq 2x^{n-1} + x^{n-2} \end{aligned}$$

We want this to be $\leq x^n$. So we have

$$2x^{n-1} + x^{n-2} = x^n \implies 2x + 1 = x^2$$

In the previous problem, you should have found r to be the root of some quadratic equation. Let s be the **other solution** to that quadratic. An explicit definition for $P(n)$ looks like $P(n) = cr^n + ds^n$ for some $c, d \in \mathbb{R}$.

Question 4c. Using the base cases of $P(n)$, determine the values of c and d and hence find an explicit definition for $P(n)$.

Note $s = 1 - \sqrt{2}$. From the assumption, we have

$$P(0) = 0 = cr^0 + ds^0,$$

so $c = -d$. We also have

$$P(1) = 1 = c(1 + \sqrt{2}) + d(1 - \sqrt{2}),$$

Solving the system of linear equations, we find that $c = \frac{1}{2\sqrt{2}}$, and $d = -\frac{1}{2\sqrt{2}}$. Thus,

$$P(n) = \frac{(1+\sqrt{2})^n - (1-\sqrt{2})^n}{2\sqrt{2}}$$

Question 4d. What is the absolute difference between $\frac{P(9)-P(8)}{P(8)}$ and $\sqrt{2}$ to 3 significant figures.

0.00000212

Question 4e. Argue informally that $\frac{P(n)-P(n-1)}{P(n-1)}$ is a 'good' approximation for $\sqrt{2}$, and that the approximation is better as n increases.

You don't need to show this, but it can be shown that this approximation is in some sense optimal!

Note that $\frac{P(n)-P(n-1)}{P(n-1)}$ is equal to $\frac{P(n)}{P(n-1)} - 1$. Which is equal to

$$\frac{(1+\sqrt{2})^n + (1-\sqrt{2})^n}{(1+\sqrt{2})^{n-1} + (1-\sqrt{2})^{n-1}} - 1$$

Since $|1-\sqrt{2}| < 1$, $(1-\sqrt{2})^n$ goes to 0 very fast. So the expression is approximately

$$\frac{(1+\sqrt{2})^n}{(1+\sqrt{2})^{n-1}} - 1 = 1 + \sqrt{2} - 1 = \sqrt{2}$$

Note that the larger n is, the smaller $(1-\sqrt{2})^n$ is, and we get a better approximation.

5 Unbalanced Recurrences

Question 5a. Use any method you want to solve the following recurrence.

$$T(n) = n + T(n/5) + T(7n/10)$$

By 'solve the recurrence', I mean find f such that $T(n) = \Theta(f)$.

Claim. $T(n) = \Theta(n)$.

Note that $T(n) = \Omega(n)$ since $T(n) = n + T(n/5) + T(7n/10) \geq n$, since T is positive.

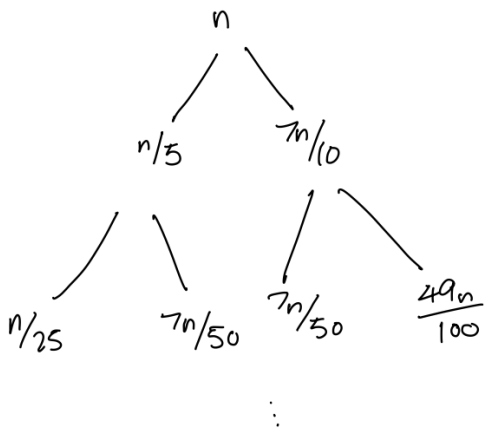
Thus, to show that $T(n) = \Theta(n)$ it suffices to show that $T(n) = O(n)$. Use the substitution method.

$$\begin{aligned} T(n) &= n + T(n/5) + T(7n/10) \\ &\leq n + cn/5 + 7cn/10 \\ &= (1 + 9c/10)n \end{aligned}$$

We need this expression to be at most cn. Solving for c we need $c \geq 10$.

Note that the recursion tree is used to get a guess, the proof that the guess is correct is done by the substitution method.

Here is how I got the guess.



total work @ this level

n

* 9/10

9/10 n

* 9/10.

81/100 n

guess work @ level h is $(9/10)^h \cdot n$

then total work = $\sum_{i=0}^{\#\text{levels}} (9/10)^h \cdot n \leq \sum_{i=0}^{\infty} (9/10)^h \cdot n$

$= n \sum_{i=0}^{\infty} (9/10)^h$

$= n \cdot \frac{1}{1-9/10}$

$= O(n)$.

So guess $O(n)$ for an upper bound.

6 Master Method Recurrences

For each of the recurrences T_i below, find f_i such that $T_i = \Theta(f_i)$. Use the master method.

Question 6a. $T_1(n) = 2T_1(n/2) + n^4$

Apply the master theorem with $a = 2, b = 2, f(n) = n^4$. Note that $n^{\log_2(2)} = n$ We have $n^4 = \Omega(n^{1+\epsilon})$, so we are in the root heavy case. Thus $T_1(n) = \Theta(n^4)$.

Question 6b. $T_2(n) = 7T_2(n/2) + n^2$

Apply the master theorem with $a = 7, b = 2, f(n) = n^2$. Note that $n^{\log_2(7)} = n^{2.81}$ We have $n^2 = O(n^{\log_2(7)-\epsilon})$, so we are in the leaf heavy case. Thus $T_2(n) = \Theta(n^{\log_2(7)})$.

Question 6c. $T_3(n) = 2T_3(n/4) + \sqrt{n}$

Apply the master theorem with $a = 2, b = 4, f(n) = \sqrt{n}$. Note that $n^{\log_4(2)} = \sqrt{n}$. We have $f(n) = \Theta(n^{\log_4(2)})$ so we are in the balanced case of the master theorem. Thus, $T_3(n) = \Theta(\sqrt{n}\log(n))$.