

1. Consider the following recurrence.

$$T(n) = T(\lfloor n/2 \rfloor) + 2T(\lfloor n/8 \rfloor) + n \quad n > 8, \text{ and } T(n) = 1 \quad 1 \leq n < 8.$$

One can prove that $T(n) = O(n)$. Do you see that $T(n) = \Omega(n)$? The goal of this problem is to show a more general statement and to refresh your induction skills.

Let c_1, c_2, c_3 be rational numbers such that $0 < c_1 \leq c_2 \leq c_3$ and $c_1 + c_2 + c_3 < 1$. Let $\alpha > 0$. Consider the recurrence

$$T(n) = T(\lfloor c_1 n \rfloor) + T(\lfloor c_2 n \rfloor) + T(\lfloor c_3 n \rfloor) + \alpha n, \quad n > 1/c_1, T(n) = 1 \quad n \leq 1/c_1.$$

- (a) Prove by induction that $T(n) = O(n)$. More precisely show that $T(n) \leq an + b$ for $n \geq 1$ where $a, b \geq 0$ are some fixed but suitably chosen constants (you get to choose and fix them based on c_1, c_2, c_3, α). You may first want to try the concrete recurrence at the start of the problem. How does a depend on c_1, c_2, c_3 ?

Solution: We proceed by induction to prove the above claim above for constants a and b to be chosen later.

For the base case of our induction, we have that $n \leq 1/c_1$. By definition, $T(n) = 1$. Thus, for our base case to hold, we require that our chosen values of a, b are such that $1 \leq an + b$ for every $n \leq 1/c_1$. Since the smallest n we can have is $n = 1$, then we simply require that $1 \leq a + b$.

For the inductive case, we have that $n > 1/c_1$. For our inductive hypothesis, suppose our claim holds for all $k < n$. Since $\lfloor c_1 n \rfloor, \lfloor c_2 n \rfloor, \lfloor c_3 n \rfloor < n$ (because $c_1, c_2, c_3 < 1$), we can apply our inductive hypothesis to each of $T(\lfloor c_1 n \rfloor), T(\lfloor c_2 n \rfloor), T(\lfloor c_3 n \rfloor)$ to obtain that

$$\begin{aligned} T(n) &= T(\lfloor c_1 n \rfloor) + T(\lfloor c_2 n \rfloor) + T(\lfloor c_3 n \rfloor) + \alpha n \\ &\leq (a\lfloor c_1 n \rfloor + b) + (a\lfloor c_2 n \rfloor + b) + (a\lfloor c_3 n \rfloor + b) + \alpha n \\ &= a(\lfloor c_1 n \rfloor + \lfloor c_2 n \rfloor + \lfloor c_3 n \rfloor) + \alpha n + 3b \\ &\leq (a(c_1 + c_2 + c_3) + \alpha)n + 3b. \end{aligned}$$

To complete our inductive step (i.e. the coefficient on n to be $\leq a$), we require that our value of a satisfy

$$a(c_1 + c_2 + c_3) + \alpha \leq a \implies \frac{\alpha}{1 - (c_1 + c_2 + c_3)} \leq a.$$

So letting a be as above and $b = 0$, we obtain that

$$T(n) \leq (a(c_1 + c_2 + c_3) + \alpha)n + 3b \leq an + b,$$

completing our inductive step. Thus, both our base case and our inductive step hold for appropriately chosen values of a (as we already set $b = 0$). Combining these restrictions, we let

$$a = \max\left(1, \frac{\alpha}{1 - (c_1 + c_2 + c_3)}\right),$$

which satisfies both conditions we set on a , completing the proof. ■

Rubric:

- 2 points: Values of a, b which work.
- 4 points: Scaled induction rubric (10 point induction rubric at the end of solutions).

- (b) Consider the recursion tree for the recurrence. What is an asymptotic upper bound on the depth of the recursion tree? Express this as a function of n and c_1, c_2, c_3 . You do not need to prove correctness of your bound.

Solution: Observe that the depth of the recurrence is upper bounded by the term given to $T(n)$ which has the slowest rate of decrease. Because $c_1 \leq c_2 \leq c_3$, the term which is decreasing slowest is $T(\lfloor c_3 n \rfloor)$, meaning that the depth of the recurrence is upper bounded by the number of times this term is recursed on. This will happen as long as the input to this term is $> 1/c_1$, so letting d be the depth of the recursion, the recursion will stop once we satisfy the following condition

$$\begin{aligned} c_3^d n &\leq 1/c_1 \\ \implies d \log_{1/c_3}(c_3) + \log_{1/c_3}(n) &\leq \log_{1/c_3}(1/c_1) \\ \implies -d &\leq \log_{1/c_3}(1/c_1) - \log_{1/c_3}(n) \\ \implies d &\geq \log_{1/c_3}(n) - \log_{1/c_3}(1/c_1), \end{aligned}$$

which bounds the depth of the recurrence. Asymptotically, this means that $d = O(\log n)$. ■

Rubric:

- 2 points: Any answer that is $\Theta(\log n)$.

- (c) We now consider a somewhat more general setting. Let $0 < c_1 \leq c_2 \leq \dots \leq c_k < 1$ be k rationals such that $\sum_{i=1}^k c_i < 1$. And $\alpha > 0$. Suppose we have a recurrence of the form

$$T(n) = \sum_{i=1}^k T(\lfloor c_i n \rfloor) + \alpha n, \quad n \geq 1/c_1, \quad T(n) = 1 \quad n \leq 1/c_1.$$

You can show that $T(n) = O(n)$ via induction as in the simpler case when $k = 3$. State the bound for a in this more general setting and also the depth of the recursion as a function of n, c_1, c_2, \dots, c_k .

Solution: The induction proof proceeds similarly to part (a), except in the step where we choose the value of a in the induction step. We now require that

$$a \left(\sum_{i=1}^k c_i \right) + \alpha \leq a \implies \frac{\alpha}{1 - \sum_{i=1}^k c_i} \leq a,$$

so we set our value of a as

$$\max \left(1, \frac{\alpha}{1 - \sum_{i=1}^k c_i} \right) = a.$$

To show our bound on the recursion depth, observe that the slowest decreasing term in the recurrence is $T(\lfloor c_k n \rfloor)$. Repeating our computation from before, we have that

$$\begin{aligned} c_k^d n &\leq 1/c_1 \\ \implies d \log_{1/c_k}(c_k) + \log_{1/c_k}(n) &\leq \log_{1/c_k}(1/c_1) \\ \implies -d &\leq \log_{1/c_k}(1/c_1) - \log_{1/c_k}(n) \\ \implies d &\geq \log_{1/c_k}(n) - \log_{1/c_k}(1/c_1), \end{aligned}$$

which asymptotically, means that $d = O(\log n)$. ■

Rubric:

- 1 point: Values of a, b that work.
- 1 point: Depth of recursion tree (also $\Theta(\log n)$).

2. Consider the set of strings $L_1 \subseteq \{\mathbf{0}, \mathbf{1}\}^*$ defined recursively as follows:

- The string ε is in L_1 .
- For any string x in L , the string $\mathbf{0}x\mathbf{1}$ is also in L_1 .
- For any string x in L , the string $\mathbf{1}x\mathbf{0}$ is also in L_1 .
- These are the only strings in L_1 .

Let $\#(a, w)$ denote the number of times symbol a appears in string w ; for example,

$$\#(\mathbf{0}, \mathbf{101110101101011}) = 5 \quad \text{and} \quad \#(\mathbf{1}, \mathbf{101110101101011}) = 10.$$

(a) Let L_2 be the set of binary strings that have equal number of $\mathbf{0}$'s and $\mathbf{1}$'s. That is $L_2 = \{x \in \{\mathbf{0}, \mathbf{1}\}^* \mid \#(\mathbf{0}, x) = \#(\mathbf{1}, x)\}$.

Prove via induction that $L_1 \subseteq L_2$.

Solution: We induct on the length n of the string. For $n = 0$, the only string in L_1 is ε , and we know that $\varepsilon \in L_2$. This takes care of the base case. Assume, as induction hypothesis, that for all $z \in L_1$ such that $|z| < n$, $z \in L_2$.

Now suppose we have a string $y \in L_1$ such that $|y| = n$. By definition of L_1 , we have $y = \mathbf{0}x\mathbf{1}$ or $y = \mathbf{1}x\mathbf{0}$ for some $x \in L_1$. In the former case,

$$\#(\mathbf{0}, y) = \#(\mathbf{0}, x) + \#(\mathbf{0}, \mathbf{1}) + \#(\mathbf{0}, \mathbf{0}) = \#(\mathbf{0}, x) + 1$$

$$\#(\mathbf{1}, y) = \#(\mathbf{1}, x) + \#(\mathbf{1}, \mathbf{1}) + \#(\mathbf{1}, \mathbf{0}) = \#(\mathbf{1}, x) + 1$$

And the same equalities holds in the latter case. Since $|x| < |y|$ by induction hypothesis $x \in L_2$ and $\#(\mathbf{0}, x) = \#(\mathbf{1}, x)$, and so

$$\#(\mathbf{0}, y) = \#(\mathbf{0}, x) + 1 = \#(\mathbf{1}, x) + 1 = \#(\mathbf{1}, y)$$

which means $y \in L_2$, and we are done. ■

Rubric:

- 4 points: Scaled induction rubric (10 point induction rubric at the end of solutions).

- (b) Give a counterexample for the statement $L_2 \subseteq L_1$.

Solution: Consider the string $0110 \in L_2$. By definition every nonempty string in L_1 starts and ends with different symbols, but this one starts and ends with 0 , so it is not in L_1 . ■

Rubric:

- 2 points: Correct counterexample.

- (c) Now consider the set of strings $L_3 \subseteq \{0, 1\}^*$ defined recursively as follows:

- The string ε is in L_3 .
- For any string x in L , the string $0x1$ is also in L_3 .
- For any string x in L , the string $1x0$ is also in L_3 .
- For any strings x and y in L , the string xy is also in L_3 .
- These are the only strings in L_3 .

Prove by induction that $L_2 \subseteq L_3$. You don't have to prove it for submission but you can generalize the first part to prove that $L_3 \subseteq L_2$ and hence $L_2 = L_3$.

Solution: We induct on length like in part a). The base case $n = 0$ only has $\varepsilon \in L_2$, which is in L_3 . Assume, as induction hypothesis, that $z \in L_3$ for any $z \in L_2$ such that $|z| < n$

Consider $s \in L_2$ of length n . We will consider cases for the first and last character of s . Suppose they are distinct, that is $s = 0x1$ or $1x0$ for some $x \in \{0, 1\}^*$. $s \in L_2$ means that s has equal number of 0 s and 1 s. By a calculation like in part a), we get x also has equal number of 0 s and 1 s. Hence $x \in L_2$. So $x \in L_3$ since $|x| < |s|$ and we can apply the induction hypothesis. But then by axiom 2 or 3 of L_3 we get $s \in L_3$.

Now suppose the first and last characters of s are the same, so $s = 0x0$ or $1x1$.

Claim 1. s can be written as wz where $w, z \in L_2$ and $|w| < |s|$ and $|z| < |s|$

Assume the claim is true, then we will have $w, z \in L_3$ by induction hypothesis. Thus $s \in L_3$ by the 4th axiom of L_3 , and we are done.

Now we will prove the claim. Let $s[i, j]$ be the substring of s from symbol i to j , including i but not j . Define a function $f : \{1, \dots, n\} \rightarrow \mathbb{Z}$ via

$$f(k) = \#(0, s[0, k]) - \#(1, s[0, k])$$

In other words, $f(k)$ is the difference between the number of 0 's and 1 's in the first k characters of s . Note that since $s \in L_2$, we have $f(n) = 0$. Also, $f(k)$ and $f(k+1)$ differ by exactly 1 for all k , since $f(k+1) = f(k) + 1$ when the k -th character is 0 , and $f(k+1) = f(k) - 1$ otherwise. Now take the case $s = 0x0$. Then $f(1) = 1$, and $f(n-1) = -1$ since $f(n) = 0$. Similarly when $s = 1x1$ we get $f(1) = -1$ and $f(n-1) = 1$. Each of those implies that $f(k) = 0$ for some $1 < k < n$, which means $s[0, k] \in L_2$. But then $s[k, n] \in L_2$ as well, since $s \in L_2$. Since $1 < k < n$ we have proved the claim with $w = s[0, k]$ and $z = s[k, n]$. ■

Rubric:

- 4 points: Scaled induction rubric (10 point induction rubric at the end of solutions).

Rubric (induction): For problems worth 10 points:

- + 1 for explicitly considering an *arbitrary* object
- + 2 for a valid **strong** induction hypothesis
 - **Deadly Sin!** Automatic zero for stating a weak induction hypothesis, unless the rest of the proof is *perfect*.
- + 2 for explicit exhaustive case analysis
 - No credit here if the case analysis omits an infinite number of objects. (For example: all odd-length palindromes.)
 - −1 if the case analysis omits a finite number of objects. (For example: the empty string.)
 - −1 for making the reader infer the case conditions. Spell them out!
 - No penalty if cases overlap (for example:
- + 1 for cases that do not invoke the inductive hypothesis (“base cases”)
 - No credit here if one or more “base cases” are missing.
- + 2 for correctly applying the *stated* inductive hypothesis
 - No credit here for applying a *different* inductive hypothesis, even if that different inductive hypothesis would be valid.
- + 2 for other details in cases that invoke the inductive hypothesis (“inductive cases”)
 - No credit here if one or more “inductive cases” are missing.