CMPS 101
Algorithms and Abstract Data Types
Fall 2014
Midterm Exam 1

Solutions

1. (20 Points) Determine whether the following statements are true or false. Give a proof if the statement is true, give a counter-example if the statement is false.

   a. (10 Points) $n\ln(n) = o(n^2)$.

   Solution: The statement is **true**.
   Proof:
   Using l'Hopital's rule we have

$$\lim_{n\to\infty}\left(\frac{n\ln(n)}{n^2}\right) = \lim_{n\to\infty}\left(\frac{\ln(n)}{n}\right) = \lim_{n\to\infty}\left(\frac{(1/n)}{1}\right) = \lim_{n\to\infty}\left(\frac{1}{n}\right) = 0$$

   showing that $n\ln(n) = o(n^2)$. ///

   b. (10 Points) If $f(n) = \Theta(n)$ and $g(n) = \Theta(n^2)$, then $f(n)\cdot g(n) = \Theta(n^3)$.

   Solution: The statement is **true**.
   Proof:
   By hypothesis there exist positive constants n_1, n_2, a_1, b_1, a_2, and b_2 such that

$$\forall\, n \ge n_1: \quad 0 \le a_1 n \le f(n) \le b_1 n$$

and

$$\forall\, n \ge n_2: \quad 0 \le a_2 n^2 \le g(n) \le b_2 n^2$$

   If $n \ge n_0 = \max(n_1, n_2)$, then both inequalities hold. Let $c = a_1 a_2$, and $d = b_1 b_2$. Since everything in sight is non-negative, we can multiply these two inequalities to get

$$\forall\, n \ge n_0: \quad 0 \le cn^3 \le f(n)g(n) \le dn^3,$$

   and hence $f(n)\cdot g(n) = \Theta(n^3)$ as claimed. ///

2. (20 Points) Use Stirling's formula to prove that $\log(n!) = \Theta\big(n\log(n)\big)$.

Proof:
Taking log (any base) of both sides of Stirling's formula, and using the laws of logarithms, we get

$$\begin{aligned}\log(n!) &= \log\left(\sqrt{2\pi n}\cdot\left(\frac{n}{e}\right)^n\cdot\big(1+\Theta(1/n)\big)\right)\\ &= \log\sqrt{2\pi n}+\log\left(\frac{n}{e}\right)^n+\log\big(1+\Theta(1/n)\big)\\ &= \frac{1}{2}\log(2\pi)+\frac{1}{2}\log(n)+n\log(n)-n\log(e)+\log\big(1+\Theta(1/n)\big).\end{aligned}$$

Therefore

$$\frac{\log(n!)}{n\log(n)} = \frac{\log(2\pi)}{2n\log(n)} + \frac{1}{2n} + 1 - \frac{\log(e)}{\log(n)} + \frac{\log\big(1+\Theta(1/n)\big)}{n\log(n)},$$

hence $\lim_{n\to\infty}\left(\frac{\log(n!)}{n\log(n)}\right)=1$ and $\log(n!)=\Theta(n\log(n))$ as claimed. ///

3. (20 Points) Consider the following algorithm that wastes time.

<u>WasteTime(*n*)</u> (pre: $n \geq 1$)
1. if $n = 1$
2. waste 1 unit of time
3. else
4. WasteTime$\big(\lceil n/2\rceil\big)$
5. WasteTime$\big(\lfloor n/2\rfloor\big)$
6. waste 3 units of time

a. (10 Points) Write a recurrence relation for the number of units of time $T(n)$ wasted by this algorithm.
Solution:

$$T(n)=\begin{cases} 1 & n=1\\ T\big(\lceil n/2\rceil\big)+T\big(\lfloor n/2\rfloor\big)+3 & n\geq 2\end{cases}$$

b. (10 Points) Show that $T(n) = 4n-3$ is the solution to this recurrence. (Hint: you may use without proof the fact that $\lfloor n/2\rfloor+\lceil n/2\rceil = n$.)
Proof:
First observe that if $T(n) = 4n-3$, then $T(1)=4-3=1$. If $n\geq 2$ then

$$\begin{aligned}\text{RHS} &= T\big(\lceil n/2\rceil\big)+T\big(\lfloor n/2\rfloor\big)+3\\ &= \big(4\lceil n/2\rceil-3\big)+\big(4\lfloor n/2\rfloor-3\big)+3\\ &= 4\big(\lceil n/2\rceil+\lfloor n/2\rfloor\big)-3\\ &= 4n-3 = T(n) = \text{LHS},\end{aligned}$$

showing that $T(n) = 4n-3$ solves the recurrence. ///

4. (20 Points) Prove that for all $n \geq 1$: if T is a tree on n vertices, then T contains $n-1$ edges. (Hint: you may use the following fact without proof. If an edge is removed from a tree, the resulting graph has two connected components.)

Proof:
Base step
If T has just one vertex, then it can have no edges, since in the definition of a graph, each edge must have distinct end vertices. Therefore $P(1)$ holds.

Induction Step (IId)
Let $n > 1$ be chosen arbitrarily, and assume for all k in the range $1 \leq k < n$, that $P(k)$ is true, i.e. for any such k, all trees on k vertices contain $k-1$ edges. We must show that $P(n)$ is true, i.e. if T is a tree on n vertices, then T has $n-1$ edges.

Let T be a tree with n vertices. Pick any edge e in T and remove it. By the hint, the removal of e splits T into two components, each of which must be tree (since no cycles are created by removing an edge) having fewer than n vertices. Suppose the two subtrees have k_1 and k_2 vertices, respectively. Since no vertices were removed, we must have $k_1 + k_2 = n$. By our inductive hypothesis, these two subtrees have $k_1 - 1$ and $k_2 - 1$ edges, respectively. Upon replacing the edge e, we see that the number of edges originally in T must have been $(k_1 - 1) + (k_2 - 1) + 1 = k_1 + k_2 - 1 = n - 1$, as required.

By the second principle of mathematical induction, all trees on n vertices have $n-1$ edges. ///

5. (20 Points) Let $T(n)$ be defined by the following recurrence relation.

$$T(n) = \begin{cases} 5 & n = 1 \\ T(\lfloor n/2 \rfloor) + 3 & n \geq 2 \end{cases}$$

a. (10 Points) Determine the values $T(2)$, $T(3)$, $T(4)$, and $T(5)$.

Solution: $T(2) = T(1) + 3 = 5 + 3 = 8$

$T(3) = T(1) + 3 = 8$

$T(4) = T(2) + 3 = 8 + 3 = 11$

$T(5) = T(2) + 3 = 11$

b. (10 Points) Prove that $T(n) \leq 8 \cdot \lg(n)$ for all $n \geq 2$. (Hint: use strong induction with two base cases.)

Proof:

I. Two base cases:

From part (a) we have $T(2) = 8 = 8 \cdot 1 = 8 \cdot \lg(2)$, and $T(3) = 8 \leq 8 \cdot \lg(3)$, so the first two cases of the inequality are true.

II. Strong Induction: $\forall n \geq 4 : \left(\forall k \in [2, n) : T(k) \leq 8\lg(k) \right) \rightarrow T(n) \leq 8\lg(n)$

Pick $n \geq 4$ arbitrarily. Assume for all integers k in the range $2 \leq k < n$ that $T(k) \leq 8\lg(k)$. In particular for $k = \lfloor n/2 \rfloor$ we have $T(\lfloor n/2 \rfloor) \leq 8\lg(\lfloor n/2 \rfloor)$. Then

$$\begin{aligned}
T(n) &= T(\lfloor n/2 \rfloor) + 3 && \text{by the recurrence formula} \\
&\leq 8\lg(\lfloor n/2 \rfloor) + 3 && \text{by the induction hypothesis} \\
&\leq 8\lg(n/2) + 3 && \text{since } \lfloor x \rfloor \leq x \\
&= 8\left(\lg(n) - \lg(2) \right) + 3 && \text{using laws of logarithms} \\
&= 8\lg(n) - 8 + 3 \\
&= 8\lg(n) - 5 \\
&\leq 8\lg(n)
\end{aligned}$$

The result follows for all $n \geq 2$ by induction. ///