CSE 431/531 Midterm Exam – Spring 2007

Time: 6:30pm to 7:50am
Place: Knox 109

Thursday Mar 08, 2007

There are totally 6 problems. You are given 2 points for writing down your name and person number correctly. In total, the maximum score is 100. There are totally 6 pages, plus one extra blank page.

Please put down your pen when you are told to do so. We shall not accept your submission otherwise.

Name	
Person Number	

Problem Number	Score obtained
name and id (2 max)	
Problem 1 (14 max)	
Problem 2 (12 max)	
Problem 3 (15 max)	
Problem 4 (20 max)	
Problem 5 (20 max)	
Problem 6 (17 max)	
Total Score: (100 max)	

Master theorem reminder: Let $a \geq 1$, $b > 1$ be constants. Suppose $T(n) = aT(n/b) + f(n)$. Then,
1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ for all sufficiently large n, then $T(n) = \Theta(f(n))$

Problem 1 (14 points)**.** Every question below has one of the following answer; answer each with a letter from **A** to **I**.

A: $\Theta(1)$, **B**: $\Theta(\lg n)$, **C**: $\Theta(n)$, **D**: $\Theta(n \lg n)$, **E**: $\Theta(n^2)$, **F**: $\Theta(n^2 \lg\lg n)$, **G**: $\Theta(n^2 \lg n)$, **H**: $\Theta(2^n)$,
I: none of the above

1. _______ is the order of $f(n) = 2^{7/n} \cdot 2^{n/7}$
2. _______ is the order of $f(n) = \lg(n!)/n$
3. _______ is the order of $f(n) = (1 + 1/n)^n$
4. _______ is the order of $f(n) = (1 + 1/1000)^n$
5. _______ is the order of $f(n) = (1 + 1/n)^{1000}$ is
6. _______ is the solution to $T(n) = nT(\sqrt{n}) + n^2$
7. _______ is the solution to $T(n) = 8T(n/3) + n^2$

Problem 2 (12 points)**.** In the following questions, mark the correct choice.

1. ☐ $2^{\log_3 n} = o(3^{\log_2 n})$
   ☐ $2^{\log_3 n} = \omega(3^{\log_2 n})$
   ☐ $2^{\log_3 n} = \Theta(3^{\log_2 n})$
   ☐ None of the above

2. ☐ $\lg^*(2^n) = o\left(2^{\lg^* n}\right)$
   ☐ $\lg^*(2^n) = \omega\left(2^{\lg^* n}\right)$
   ☐ $\lg^*(2^n) = \Theta\left(2^{\lg^* n}\right)$
   ☐ None of the above

3. ☐ $2^{\sqrt{n}} = o\left(\sqrt{2}^n\right)$
   ☐ $2^{\sqrt{n}} = \omega\left(\sqrt{2}^n\right)$
   ☐ $2^{\sqrt{n}} = \Theta\left(\sqrt{2}^n\right)$
   ☐ None of the above

4. ☐ $\lceil \lg n \rceil ! = o\left(\lg(n!)\right)$
   ☐ $\lceil \lg n \rceil ! = \omega\left(\lg(n!)\right)$
   ☐ $\lceil \lg n \rceil ! = \Theta\left(\lg(n!)\right)$
   ☐ None of the above

5. ☐ $3^{\frac{1}{\sqrt{n}}} = o\left(\sqrt{3}\right)$
   ☐ $3^{\frac{1}{\sqrt{n}}} = \omega\left(\sqrt{3}\right)$
   ☐ $3^{\frac{1}{\sqrt{n}}} = \Theta\left(\sqrt{3}\right)$
   ☐ None of the above

6. ☐ $\frac{n}{\lg n} = o\left(\frac{\lg n}{\lg\lg n}\right)$
   ☐ $\frac{n}{\lg n} = \omega\left(\frac{\lg n}{\lg\lg n}\right)$
   ☐ $\frac{n}{\lg n} = \Theta\left(\frac{\lg n}{\lg\lg n}\right)$
   ☐ None of the above

Problem 3 (15 points). TRUE or FALSE? If you choose FALSE, give a counter example to briefly justify the choice; otherwise, you don't have to justify your answer. If a question involves functions f and g, implicitly assume that $f, g : \mathbb{N} \to \mathbb{R}^+$, and that $f(n), g(n) \to \infty$ as $n \to \infty$.

1. Given any edge-weighted undirected graph $G = (V, E)$ and a source vertex s, any shortest path tree (with source s) of G is also a minimum spanning tree of G.

   □ TRUE

   □ FALSE

2. Consider an edge-weighted undirected graph $G = (V, E)$ and a source vertex s. Let T be a shortest path tree (with source s). Suppose we add 1 to the weight of each edge of G. Then, T remains a shortest path tree with respect to the new weights.

   □ TRUE

   □ FALSE

3. Consider an edge-weighted undirected graph $G = (V, E)$. Let T be a minimum spanning tree. Suppose we add 1 to the weight of each edge of G. Then, T remains a minimum spanning tree with respect to the new weights.

   □ TRUE

   □ FALSE

4. $f(n) = \Omega(g(n))$ implies $f(n) = \omega\left(\frac{g(n)}{2}\right)$

   □ TRUE

   □ FALSE

5. $f(n) = O(g(n))$ implies $f(n) = o\left((g(n))^2\right)$

   □ TRUE

   □ FALSE

Problem 4 (20 points)**.** In the MERGE-SORT algorithm, we have a MERGE() sub-routine which can merge any two sorted sub-arrays of lengths x and y in time $O(x + y)$. Suppose we have m sorted sub-arrays $A_1, \ldots, A_m$ of length n each, and we would like to merge them, using MERGE(), into a sorted array of length mn.

(a) One way to do this "multiway merge" is to merge A_1 with A_2, then merge the result with A_3, whose result is merged with A_4 and so on. What is the running time of this algorithm? Justify your answer.

(b) Devise an asymptotically faster algorithm to merge these m sub-arrays. Describe your idea and analyze its running time. No pseudo-code is needed.

Problem 5 (20 points)**.** In Viet Nam, school students are punished every day for various violations of the Student Code of Conduct. A punishment as old as time itself is to write down many times some slogan such as "*I shall never knock unconscious any of my classmates again.*" The number of times a student has to write a slogan is dependent on the severity of the violation. (Do not ask me how many slogans I've written down in my time!)

One day, there is a pool of n troubled kids. Let's number them from 1 to n for the sake of their anonymity and privacy. Kid number i is supposed to write some slogan t_i times. The school principal is quite cruel, but he is not very algorithmically smart. He has a set of n favorite slogans, where the jth slogan takes s_j seconds to be written down. He wants to assign to each kid a *distinct* slogan from this set. At the same time, he also wants to maximize the collective amount of time the poor kids have to pay for their bad behavior.

For example, suppose $n = 3, t_1 = 1, t_2 = 2, t_3 = 3, s_1 = 5, s_2 = 2, s_3 = 4$. If kid 1 gets slogan 2, kid 2 gets slogan 1, and kid 3 gets slogan 3, then the total time spent is $t_1 s_2 + t_2 s_1 + t_3 s_3 = 1 \cdot 2 + 2 \cdot 5 + 3 \cdot 4 = 24$ seconds.

(i) In words, briefly describe an algorithm to solve this SLOGAN ASSIGNMENT problem for the principal.

(ii) Prove your algorithm's correctness.

Problem 6 (17 points)**.** Consider a connected graph $G = (V, E)$ whose edges have distinct costs and $|E| \geq |V|$. Suppose you are given the minimum spanning tree T_1 of G, and the second cheapest spanning tree T_2 of G. Show that there is an edge $e_1 \in T_1$ and an edge $e_2 \in T_2$ such that $T_1 \cup \{e_2\} - \{e_1\} = T_2$.

Note for this particular problem:

- The solution is short. If you're writing too many words, something is wrong.
- To reward academic honesty: if your solution makes no sense, 4 points will be deducted. If you don't answer the question or admit that you don't know the answer, you'll be given 4 points for free.)