Midterm Examination on Algorithms Nov. 25, 1998

Teacher: Biing-Feng Wang

Problem 1: (10%, Recurrences)

(1) (4%) Let $T(n)=6T(n/2)+ n$. Find the tight asymptotic bound of $T(n)$ in Q-notation by appealing to a recursion tree.

(2) (3%) $T(n)=2T(n/2) + n4/5$. Find the tight asymptotic bound of $T(n)$ in Q-notation by appealing to the master theorem.

(3) (3%) $T(n)=9T(n/3) + n2$. Find the tight asymptotic bound of $T(n)$ in Q-notation by appealing to the master theorem

Problem 2: (15%, Growth of Functions)

Let $f(n)$ and $g(n)$ be asympotically positive functions. Determine whether each of the following statements is correct or not.

(1) $f(n) =O(g(n))$ implise $g(n)=O(f(n))$.

(2) $f(n) + g(n)=Q(min\{f(n), g(n)\})$.

(3) $f(n)=O(g(n))$ implies $lg(f(n))=O(lg(g(n)))$, where $lg(g(n))>0$ and $f(n)*1$ for all sufficiently larger n.

(4) $f(n)=O(g(n))$ implies $2f(n)=O(2g(n))$.

(5) $f(n)=O(g(n))$ implies $g(n)=W(f(n))$.

(6) $f(n)=Q(f(n/2))$.

(7) $f(n) + o(f(n)) =Q(f(n))$.

Problem 3: (10%, Heap)

Show that Building a heap of size n can be done in linear time.

Problem 4: (20%, Sorting)

(1) (10%) Prove that W(nlog n) is a time lower bound for sorting in the comparison model.

(2) (5%) Name three sorting algorithms that are optimal in the comparison model and three algorithms that are not optimal in the comparison model.

(3) (5%) Let A[1..n] be an array storing all the telephone numbers of the people in Hsinchu. Give an efficient algorithm to sort the elements of A. What's the time complexity of your algorithms.

Problem 5: (15%, Divide-and-Conquer)

Explain and compare the following three algorithm design strategies:

divide-and-conquer, partition, prune-and-search.

Problem 6: (10%, Greedy Algorithms)

Professor Midas drives a car from Newark to Reno along the road Inter-state 80. His car's gas tank, when full, holds enough gas to travel n miles, and his map gives the distances between gas stations on his route. The professor wishes to make as few gas stops as possible along the way. Give an efficient method to determine which gas stations he should stop, and prove that your strategy yields an optimal solutin.

Problem 7: (10%, Dynamic Programming)

(1) (3%) Let X[1..m] and Y[1..n] be two strings. Let c[i, j] be the length of the LCS (longest common subsequence) of X[i..m] and Y[j..n]. Write a recurrence of c[i, j].

(Note that the definition of c[i, j] is not the same with that in the textbook.)

(2) (4%) Write an algorithm to compute the length of the LCS of X[1..m] and Y[1..n].

(3) (3%) What is the time complexity of your algorithm?

Problem 8: (10%, Medians and Order Statistics)

The k-th quantiles of an n-element set S are the k-1 order statistics that divide the sorted set into k equal-sized sets. (Assume that n is a multiple of k.) For example, let k=3, n=9, and S={15, 26, 63, 38, 81, 97, 52, 43, 76}. The k-th quantiles of S is (38, 63), since 38 and 63 are the 3-th and 6-th smallest elements in S, respectively, and thus they can be used to divide S into the following three equal-sized sets:

{15, 26, 38}, {43, 52, 63}, {76, 81, 97}.

Give an O(nlog k)-time algorithm to find the k-th quantiles of an n-element set S.

Bonus: (10%, Elementary Graph Algorithms)

Let T be a rooted tree with root r. Each edge e in T is associated with a weight w(e). The length of a path in T is the total weight of the edges in the path. Let v be a node in T. We define the bisector nodes of v as follows. Let (r=p0, p1, p2, ..., pk=v) be the unique path from r to v. Let m be the midpoint of the path. If m is located at a node pq, the node pq is the unique bisector node of v. Otherwise, m is located at an edge (pq-1, pq) and both nodes pq-1 and pq are bisector nodes of v. Given the rooted tree T, the bisector problem is to find the bisectors nodes of every node in T. For example, let T be given as the following, the output should be:

r: {r} a: {r, a} b: {r, b} c: {a, c} d: {a} e: {b}.

EMBED MSDraw * mergeformat

(1) (5%) Give a polynomial time algorithm for the bisector problem. Determine the time complexity of your algorithm (in big-O notation).

(2) (5%) Redo (1) under the assumption that the weight of every edge e in T is 1 (i.e., T is unweighted).

Roll Call: (x%, Rabbit and Frog)

在慈悲的兔子與可憐的青蛙之故事中，最後那隻青蛙怎麼了？

(註: 這個故事可以很好笑，只要你把那隻青蛙想像成是教授就可以了。)