Final Examination on Algorithms

Teacher: Biing-Feng Wang **June 13, 2002**

Problem 1: (10%, problem from midterm exam -- growth of functions)

(1) (5%) Give the definition of $\Theta(g(n))$.

(2) (5%) Prove that $9n^3-4n=\Theta(n^3)$ according to the definition in (1).

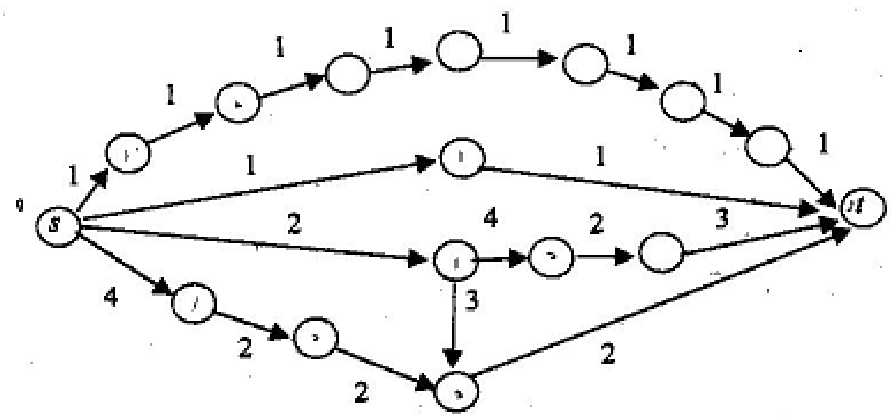
Problem 2: (10%, problem from midterm exam -- sorting) Give an $o(n\log n)$-time algorithm for sorting n integers in the range $[1..n^{20}]$.

Problem 3: (10%, problem from midterm exam -- dynamic programming) For any non-decreasing sequence of numbers $S=(s_1, s_2, \ldots, s_c)$, define $cost(S)=(s_c-s_1)^2$. If $|S|=0$ or $|S|=1$, we define $cost(S)=0$. Let $A=(a_1, a_2, \ldots, a_n)$ be a non-decreasing sequence of n positive numbers and k be an integer, where $1\le k\le n$. In this problem, you are asked to partition A into k sub-sequences $A_1, A_2, \ldots, A_k$ such that $\max_{1\le i\le k}\{cost(A_i)\}$ is minimized. Give an efficient algorithm to compute the minimum value of $\max_{1\le i\le k}\{cost(A_i)\}$. What is the time complexity of your algorithm?

Problem 4: (15%, maximum flow)

(1) (5%) The Ford-Fulkerson method solves the maximum flow problem on a graph G by repeatedly finding an augmenting path p in the residual network G_f and then augmenting the flow f along p. Why do we call it a method instead of an algorithm?

(2) (10%) Find the maximum flow of the following graph by using Edmonds-Karp algorithm. Please illustrate f, G_f, and p step by step.



94u320

Problem 5: (13%, approximation algorithms)

(1) (4%) Give an approximation algorithm for the vertex-cover problem.

(2) (4%) What is the time complexity of your algorithm? Please explain.

(3) (5%) Prove the error bound of your algorithm.

Problem 6: (10%, convex hull)

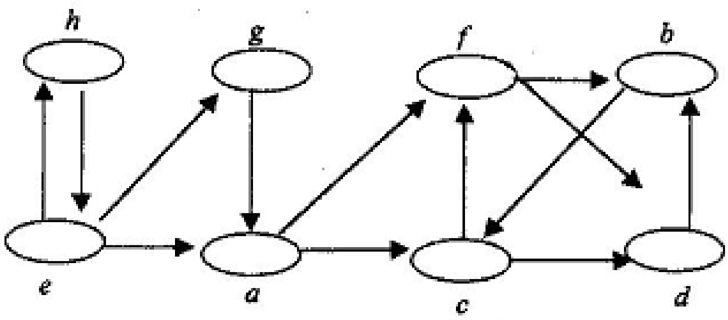
(1) (5%) Describe Graham's scan.

(2) (5%) What is the time complexity of Graham's scan? Please explain.

Problem 7: (10%, divide-and-conquer) Let A and B be two integers of n bits. Give an $o(n^2)$-time divide-and-conquer algorithm to compute $A \times B$. What is the running time (in bit-complexity) ?

Problem 8: (12%, all-pairs shortest paths) Give an $O(n^3)$ time algorithm for computing the transitive closure of a graph G, using a storage of $O(n^2)$ bits.

Problem 9: (10%, DFS) Illustrate *depth-first forest* by performing DFS on the following graph. Assume that the vertices in the adjacency list of each vertex are in alphabetical order. Therefore, in the following graph, while f is visited, we check b first and then check d. Also assume that in case a new *root* is needed, the vertices are chosen in alphabetical order. Therefore, the root of the first search tree is a. You should label each node v by $d[v]/f[v]$ and classify each edge into tree edge, cross edge, back edge, or forward edge, where $d[v]/f[v]$ denotes the *discovered/finished* time.



Bonus: (10%, A problem selected from homework) Suppose we label the nodes of binomial tree B_k in binary by a postorder walk. Consider a node x labeled at depth i, and let $j=k-i$. Show that x has j 1's in its binary representation.