Data Structures and Algorithms: Jan—Apr 2015

Final Exam

Instructions. There are two parts with four questions each. The total weightage is 50 marks. When algorithms are asked, give a brief, but complete, description (in pseudocode form or in the form of steps) and a brief (in one or two sentences) proof of correctness too.

Part A ($4x3 = 12$ marks)

1. Which of the statements is true?
   The second largest element from a sequence of n elements,
   (a) can be found using at most $n - 1 + n - 2 = 2n - 3$ comparisons, but can't be done better,
   (b) can be found using at most $n - 1 + \lceil \log n \rceil - 1$ comparisons, but it is not clear that it is optimal,
   (c) can be found using at most $n - 1 + \lceil \log n \rceil - 1$ comparisons, and there is a matching adversary based lower bound for the worst case,
   (d) can be found using at most $n-1+\lceil \log n \rceil -1$ comparisons, and any algorithm needs this many comparisons for any sequence of inputs.

2. Which of the following statements is false?
   (a) Given an undirected graph and an integer k, it is NP-complete to determine whether there is a cycle of length at least k in the graph.
   (b) Given an undirected graph and an integer k, it is NP-complete to determine whether there is a cycle of length at most k in the graph.
   (c) Given an undirected graph, finding a partition of the vertex set into two such that minimum number of edges go between the two sets (Minimum Cut), can be done in polynomial time.

3. Which of the following does NOT have an algorithm that takes time linear in the input size?
   (a) Testing whether a given undirected graph is connected
   (b) Multiplying two polynomials each of degree n
   (c) Finding the median of a list of n numbers

4. We are required to maintain a set of n numbers under insert, delete and findmin operations. Which of the following is true?
   (a) A heap can do all these operations in $O(\log n)$ time and using just an array.
   (b) If the position of the element to be deleted is not given with the query, then a heap can not perform this in $O(\log n)$ time, and a binary search tree can do all these in $O(\log n)$ time.
   (c) A heap can not perform this in $O(\log n)$ time even if the position of the element to be deleted is given.

Part B (First three questions carry 9 marks each, the last one carries 11 marks)

1. Show **ONE** of the two problems is NP-complete. You can assume any of the problems we have shown in class (SAT, Vertex Cover, Clique, Independent Set, 3-Coloring, Hamiltonicity, Feedback Vertex Set....) are NP-complete.
   (a) **Subgraph Isomorphism** Given two undirected graphs G and H, determine whether there is a subgraph of G isomorphic to H. Two graphs G_1 and G_2 are isomorphic if there is a bijection f between the vertex set of G_1 and the vertex set of G_2 such that (u, v) is an edge in G_1 if and only $(f(u), f(v))$ is an edge in G_2.

(b) **k-coloring** Consider the k-coloring problem where given an undirected graph, we would like to determine if its vertex set can be colored with at most k colors such that adjacent vertices get different colors. Show that k-coloring problem, for a fixed $k > 3$ is NP-complete assuming that 3-coloring is NP-complete.

2. Do **ONE** of the following problems.

   (a) Given a tree, give a polynomial time algorithm to find the minimum vertex cover in the graph. [Hint: What would you do with leaves of the tree?]

   (b) Give an $O(n + m)$ algorithm for the following problem. Given an undirected graph on n vertices and m edges given by its adjacency list, determine whether it is bipartite.

3. Do **ONE** of the following problems.

   (a) Give an $O(n)$ time algorithm to determine whether a given graph (in the form of an adjacency list) on n vertices and m edges is a tree. Note that your algorithm should take time independent of the number m of edges in the graph, and you are not given m. You can infer it though, if necessary, by finding the degree of every vertex.

   (b) Let $M(n)$ be the time to multiply two n by n matrices. Given an undirected graph, show that one can determine in $M(n)$ time whether the graph contains a triangle (beating the naive $O(n^3)$ time).

4. Do **ONE** of the following problems.

   (a) You have decided to make a CD of your favorite songs. You already picked the songs, but you still need to decide in what order they should appear on the CD. Your task is to compute the "best" possible order for them to appear. For each pair of songs i and j you have chosen a "compatibility rating" $compat_{i,j}$, which is a positive number indicating how good you think j sounds when it is played right after i. In general, $compat_{i,j}$ and $compat_{j,i}$ need not be equal.

   Use dynamic programming to find an ordering of the songs which maximizes the sum of the compatibilities of consecutive songs. You do not need to print out the actual ordering, just the maximum total compatibility. For n songs your algorithm should run in $O(n^2 2^n)$ time.

   (b) A group of traders are leaving India, and need to convert their Rupees into various international currencies. There are n traders and m currencies. Trader i has T_i Rupees to convert. The bank has B_j Rupees worth of currency j. Trader i is willing to trade as much as C_{ij} of his Rupees for currency j. (For example, a trader with 1000 Rupees might be willing to convert up to 700 of his Rupees for USD, up to 500 of his Rupees for Japaneses Yen, and up to 500 of his Rupees for Euros).

   Assuming that all traders give their requests to the bank at the same time, describe an algorithm that the bank can use to satisfy the requests (if it can). [Hint: Formulate the problem as a flow network.]