CS570
Analysis of Algorithms
Spring 2009
Exam I- Solution

1) 20 pts
Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[TRUE/FALSE]
Given a weighted graph and two nodes, it is possible to list all shortest paths between these two nodes in polynomial time.
False. Not valid for graph with negative weights.

[TRUE/FALSE]
Given a graph G = (V, E) with cost on edges and a set S which is a subset of V, let (u,v) be an edge such that (u, v) is the minimum cost edge between any vertex in S and any vertex in V - S. Then, the minimum spanning tree of G must include the edge (u, v).

True. Refer to textbook, page 145, theorem 4.17.

[TRUE/FALSE]
Let G = (V, E) be a weighted graph and let M be a minimum spanning tree of G. The path in M between any pair of vertices v1 and v2 must be a shortest path in G.

False. Counter example: a loop with three nodes A, B and C. The edge costs are A-B: 1, B-C: 1, A-C: 1.5. Edge A-C is not included in M, but the shortest path between A and C is through edge A-C.

[TRUE/FALSE]
$2n^2 + \theta(n) = \theta(n^2)$

True.

[TRUE/FALSE]
Kruskal's algorithm can fail in the presence of negative cost edges.

This is False.

[TRUE/FALSE]

For any cycle in a graph, the cheapest edge in the cycle is in a minimum spanning tree.
False. If every edge in a cycle C is very expensive then it is possible that the MST doesn't use any edge in C (including the cheapest edge in C).

[TRUE/FALSE]
In every instance of the stable marriage problem there is a stable matching containing a pair (m, w) such that m is ranked first on the preference list of w and w is ranked first on the preference list of m.

False. Consider the following scenario where there are two men and women each in M and W with the following preferences:

m prefers w to w'

m' prefers w' to w

w prefers m' to m

w' prefers m to m'

[TRUE/FALSE]
Let T(n) be a function obeying the recurrence T(n) = 5T(n/5) + a with initial condition T(1) = b, where a and b are positive numbers. Then T(n) = Θ(n log n).
False. By the Master Theorem T(n) = Θ(n). We compare $n\log_5^5$ with the overhead term of a = $O(n^0)$ and find that we are in the case where the former dominates.

[TRUE/FALSE]
If the edges in a connected undirected graph are unit cost, then you can find the MST using BFS.

This is true.
[TRUE/FALSE]
Asymptotically, a running time of n^d is better than $10n^d$

This is False. Asymptotically, both of them are the same Basically within theta of each other

2) 10 pts

Give an algorithm that given as input a sequence of n numbers and a natural number $k < n$, outputs the smallest k numbers of the sequence in O(n + k log n) time.

Solution: Building a MIN-HEAP takes O(n) time and doing EXTRACT-MIN k times takes O(k log n) time and gives us the k smallest numbers of the sequence. Hence running time is O(n + k log n).

3) 10 pts
Arrange the following in the increasing order of asymptotic growth (x is a constant which is greater than 1).

$x^{\sqrt{\log n}}$ $\quad x^n$ $\quad n^{10}$ $\quad n(\log n)^4$ $\quad n^{\log n}$ $\quad x^{x^n}$ $\quad x^{n^x}$

Solution:

$x^{\sqrt{\log n}}$ $\quad n(\log n)^4$ $\quad n^{10}$ $\quad n^{\log n}$ $\quad x^n$ $\quad x^{n^x}$ $\quad x^{x^n}$

4) 20 pts

You have a sufficient supply of coins, and your goal is to be able to pay any amount using as few coins as possible.

(a) Describe a greedy algorithm to solve the problem when the coins usedare of values 1, 5, 10 and 25 cents. Prove the optimality of the algorithm.

Solution: Let k be the number of coins of denomination d_i just enough to be greater than equal to d_{i+1}, where d_{i+1} is the next higher denomination. Then k coins of denomination d_i can be replaced by d_{i+1} and some other coin(s) using less than k coins. One can verify this observation by taking into account the fact that 5 pennies can be replaced by a nickel, 2 nickels can be replaced by a dime and 3 dimes can be replaced by a quarter and a nickel (2 coins).

Let the money to change for be n, n = 25a + 10b + c + d. a, b, c, d are the numbers of quarters, dimes, nickels and pennies. We have b < 3, c < 2 and d < 5 (or it will violate the conclusion in the observation, e.g., if b □ 3, we can replace 3 dimes by a quarter and a nickel which results in less coins). We prove that when n □ 25, we should always pick a quarter. As 10b+c+d □ 10*2+5*1+1*4 = 29, it means in the optimal solution, the amount of the money of the dimes, nickels and pennies will be no more than 29 cents. So we only need to prove the cases when n is 25, 26, 27, 28 or 29. One can easily verify that in such cases it is always better to choose a quarter. Similarly we can prove that when 10 □ n < 25, we should always pick a dime and when 5 □ n < 10 we should always pick a nickel.

(b) Show that a greedy algorithm does not yield optimal results if you are using only coins of value 1 cent, 7 cents and 10 cents.

Solution: Counter example: Consider making change for x = 14. The greedy algorithm yields 1× (10 cents) + 4× (1 cents). This is 5 coins. However, we can see that 2× (7 cents) = 14 = x also. Here we use 2 coins. Obviously 2 < 5, so Greedy is sub-optimal in the case of coin denominations 1, 7, and 10.

5) 10 pts

Let T be a minimum spanning tree for G with edge weights given by weight function w. Choose one edge (x, y) $\in$ T and a positive number k, and define the weight function w’ by

w’(u, v) = w(u, v), if (u, v) $\neq$ (x, y)

w’(u, v) = w(u, v) – k, if (u, v) = (x, y)

Show that T is also a minimum spanning tree for G with edge weights given by w’.

Solution: Proof by contradiction. Note T is a spanning tree for G with weight function w’ and w’(T) = w(T) – K.

Suppose T is not a MST for G with w’ and hence there exists a tree T’ which is an MST for G with w’. So w’(T’) < w’(T) (*).

We have two cases to analyze. First, if (x, y)) Î T’ then w’(T’) = w(T’) – k and from the above we have w(T’) < w(T). Now T’ is a spanning tree for G with w which is better than T which is a MST for G with w. A contradiction. Case 2, suppose (x,y) Ï T’ then w’(T’) = w(T’) and so by (*) we have w(T’) < w’(T) = w(T) – k. A similar contradiction again.

Hence T is a MST for G with w’.

6) 15 pts

Stable Matching: Prove that, if all boys have the same list of preferences, and all the girls have the same list preferences, there can be only one stable marriage.

Solution: Proof by contradiction. Suppose there are n boys b1, b2, . . . , bn; and n girls g1, g2, . . . , gn. Without loss of generality, assume that all the boys prefer g1 over g2, g2 over g3, and so on. Likewise all the girls most prefer b1 and least prefer bn. It is easy to verify that the pairing (b1, g1), (b2, g2), . . . , (bn, gn) is a stable matching. Suppose that there is another stable matching, including couple (bi, gj) for i != j. If there are multiple such pairs, let i be the smallest such value. Therefore j > i, because g1 to gi- 1 are engaged to b1 through bi- 1 respectively: so j cannot be less than i. By the definition of the problem, everyone is engaged, including gi, and thus for some k > i, our alternate pairing includes the couple (bk, gi). Note that bi is a more popular boy than bk, since we chose i to be the smallest possible value. Therefore gi prefers bi over her fiance. And because i < j, we know bi prefers gi over his fiancee. Thus bi and gi constitute a instable couple. But this contradicts our supposition that this alternate matching is stable. So we conclude that there is no alternate stable matching.

7) 15 pts
Give a divide-and-conquer algorithm to find the average of n real numbers. You should clearly show all steps (divide, conquer, and combine) in your pseudo code. Analyze complexity of your solution.

Solution:

Divide: divide array up in 2 equal pieces
Conquer: solve the two subproblems reursively and return the average
Combine: Compute the average for the orginal problem by:
Average = (number of items in sub problem 1 * average for subproblem 1 + number of items in sub problem 2 * average for subproblem 2)/ (size of the original problem)

Complexity of Divide is O(1), complexity of Combine is O(1). Master theorem gives you a complexity of O(n) for the solution.

Additional Space

Additional Space