UC Berkeley – CS170: Introduction to CS Theory
Professor Satish Rao

Midterm 1
March 6, 2002
Last revised March 6, 2002

Midterm 1

Note

(a) You have **2 hours** to complete this midterm.

(b) The questions are arranged roughly in increasing order of difficulty. It would be wise to finish the simpler questions before tackling the harder ones.

(c) When asked for an algorithm you must give 1) A brief informal definition of the algorithm 2) A more precise definition, possibly using pseudo-code 3) A proof of correctness 4) An analysis of the running algorithm

(d) When asked for a quantity *e.g.* "How many comparisons are required to sort n numbers" you must give both the number as well as your justification for that answer unless the question says otherwise.

(e) Best of Luck!

Problem 0

[4 points] State briefly the following:

(a) Your *name*: ____________________

(b) Your *SID number*: ____________________

(c) The section you go to (TA & time): ____________________ ____________________

(d) Your account name: ____________________

Problem 1

(a) **[10 points]** TRUE or FALSE: Just state whether the following statements are true or false. Suppose that $f(n) = 12n + 6$.

(i) $f(n)$ is $O(n^2)$.

(ii) $f(n)$ is $\Omega(n^2)$.

(iii) Karatsuba's algorithm for multiplying two n-bit numbers is asymptotically faster than $O(n^2)$. (HINT: The recurrence for Karatsuba's algorithm is $T(n) = 3T(n/2) + O(n)$ for $n > 1$ and $T(1) = O(1)$.)

(iv) The number of bits in the nth Fibonacci number is exponential.

(v) The Mayans had a positional system.

(vi) Al Gore invented algorithms.

(b) **[5 points]** What is the *expected* number of collisions when using a random hash function from 2-univeral family to hash n elements of a universe M into a table of size $2n$?

Problem 2

(a) **[10 points]** Give an example of a graph with no negative cycles, and a source vertex for the graph, where Dijkstra's algorithm *does not* compute a correct shortest path tree. (Of course the graph can have negative edges…)

(b) **[10 points]** Consider the graph shown in the figure below (Fig. 1). For this graph, give a 1) breadth first search tree, 2) a depth first search tree, 3) a shortest path tree, and 4) a minimum spanning tree.

NOTE: For the BFS tree and the DFS trees *ignore the edge weights* but consider the direction of the edges. For the MST, ignore the direction of the edges, but consider the edge weights. Wherever a starting vertex is required, let the node "s" be the starting vertex.

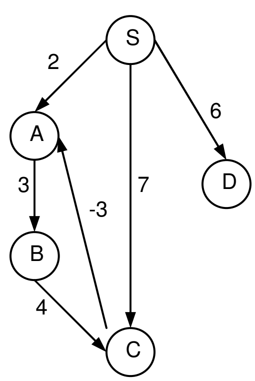

Figure 1: Graph for Problem 2

(c) **[5 points]** An *articulation point* in a graph is a node whose removal leaves more than one connected component. Give an $O(|V| \cdot |E|)$ algorithm for finding *one* articulation point.

Problem 3

TRUE or FALSE: For each of the following statements just state whether they are true or false. **NOTE:** If you are sure of your answer, you may simply say true or false, but incorrect one word answers are penalized a point. However, it is possible to get partial credit by adding a *brief* justification for your answer even if that answer is wrong.

(a) **[9 points]**
Depth First Search:

(i) If in a depth first search of a directed graph, there are no back edges, then there are no cycles.

(ii) If in depth first search of a directed graph, there are no cross edges, then there are no cycles.

(iii) In a depth first search of a directed acyclic graph, there are no edges from a node with a lower post number to a node with a higher post number.

(b) **[9 points]** Consider the distance labels assigned during a *breadth first search* of a graph.

(i) No edge connects two notes with the same distance label.

(ii) No edge connects two nodes with different distance labels.

(iii) No edge connects two nodes with distance labels that differ by more than 1.

Problem 4

[15 points]
Show that the strictly heaviest edge (assume no ties) on a cycle in a weighted graph is definitely not in *any* Minimum Spanning Tree of the graph. (HINT: Recall that removing an edge from a tree divides the tree into exactly two connected trees, and adding an edge between the trees makes it a spanning tree.)

Problem 5

[15 points]
NOTE: If you get stuck on this question, then move on to Problem 6, the first part of which is easy.

Do ONE of the following problems.

(a) Give an undirected graph $G = (V, E)$, where edges represent two way roads and the weights on the edges represent an *upper* limit on the weight of vehicles that can use that road. Give an $O(|E| \log |V|)$ time algorithm that finds the weight of the heaviest truck that can visit *every* node. HINT: Spanning Trees

OR

(b) Consider an undirected graph $G = (V, E)$. A subset $F \leq V$ of the vertices are facilities and another subset $C \leq V$ are consumers.

For each consumer we wish to find the facility closest to her, *i.e.* for each $c \in C$ we wish to find the vertex $v \in F$ that is closest to the consumer on the graph. Give an $O(V + E)$ time algorithm to do the above. HINT: Add a special vertex …

Problem 6

(a) **[7 points]** Given a graph with one negative edge, show how to determine whether there is a negative cycle in $O((V + E) \log V)$ time. (HINT: remove the negative edge (u, v) and do a shortest path computation using v as the source.)

(b) **[10 points]** Give a shortest path algorithm for a graph with k negative edges that runs in $O(k(V + E) \log V)$ time. (HINT: suppose there was one negative edge (u, v). Find a good price function, and compute the shortest path. Now generalize to k negative edges…)