~

CS170 First Midterm 23 Sep 1999

NAME:__________________________

TA:______________________________

Key (see below):______________________

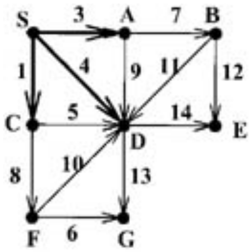
Be clear and concise. You may use the number of points assigned to each problem as a rough estimate for the number of minutes you want to allocate to the problem. The total number of points is 80.

This exam is closed book/closed computer, but you are allowed one sheet of paper (2 sides) for notes to use during the exam. You are allowed to use, without proof, any results proven in class, lecture notes, homework, or the book, but you must state clearly what result you are using.

We are no longer allowed to post grades using SIDs, because of privacy rulings. So we ask you to pick a key that we will use to post your grades on-line. Pick something you can remember, that does not identify you personally, and that is very likely to be unique. Try adding the last 3 digits of your SID to guarantee uniqueness.

1. (**15 points**) We are running one of these three algorithms on the graph below, where the algorithm has already processed the bold-face edges. (Ignore the directions on the edges for Prims and Kruskals algorithms.)

- •Prims for the minimum spanning tree, starting from S.
- •Kruskals for the minimum spanning tree.
- •Dijkstras for shortest paths from S.



Which two edges would be added next in Prims algorithm? Be sure to indicate the order in which they are added.

Which two edges would be added next in Kruskals algorithm? Be sure to indicate the order in which they are added.

At this point in the running of Dijkstras algorithm, S has been taken off the top of the heap and marked. Which four vertices would be marked next in Dijkstras algorithm, i.e. deleted from the top of the heap? Be sure to indicate the order in which they are deleted. Which final edges would Dijkstras algorithm choose as part of the shortest path to these vertices (i.e. which edge connects to this vertex as part of the shortest path from S)?

2. (**25 points**) Let G = (V, E) be a directed graph where every edge *e* has a weight $w(e)$, which may be positive, negative, or zero. The *benefit* of a path consisting of edges e_1, e_2, ... e_k is defined as $\min_{1 \le i \le k} w(e_i)$.

Give an algorithm which computes the *maximum benefit* path from a given vertex *s* to another vertex *f*. For example, if there are two paths from *s* to *f*, where Path 1 has edge weights -1, -2, -3 and Path 2 has edges weights -4, 10, 20, then Path 1 has benefit -3 and Path 2 has benefit -4, so Path 1 has the maximum benefit: -3 = max(-3,-4).

Your algorithm should be as efficient as possible.

A. Brief description or pseudocode

B. Justification of correctness.

C. Running time and justification. You may refer to analyses done in class without proof.

3. (**25 points**) A directed graph G = (V, E) is *semiconnected* if for every pair of distinct vertices u and v, there is either a path from u to v, or a path from v to u, or both (u and v lie on a cycle). Show that G is semiconnected if and only if the DAG formed by its strongly connected components has a unique topologically sorted order, i.e. there is a unique way to order the DAG vertices $v_1, v_2, ..., v_n$ such that any edge (v_i, v_j) satisfies $i < j$.

4. (**15 points**)

True or false? No explanation required. except for partial credit. Each correct answer is worth 1.5 points, but 1.5 points will be *subtracted* for each wrong answer, so answer only if you are reasonably certain.

(a) $n^{2+\sin n} = O(n^2)$

(b) Dijkstras algorithm can fail if there are zero edge weights along with positive ones.

(c) Let G be an undirected, weighted graph. Running an MST algorithm on each biconnected component of G, and then adding the bridge edges to the set of MSTs of each biconnected component, results in a combined MST for the whole graph.

(d) Let G = (N,E) be an undirected graph. Let $N = N_1 \cup N_2$ be a partition of N into nonempty disjoint subsets, and G_i the graph consisting of N_i and all edges from E with both endpoints in N_i. Let T_i be an MST of G_i. Then one can construct an MST T of G by connecting T_1 and T_2 by the shortest edge connecting N_1 and N_2.

(e) $\sum_{i=1}^{n} i^r (\log_2 i)^s = \Theta(n^{r+1} (\log_2 n)^s)$, if r > 0 and s > 0.

(f) If we add a directed edge to a directed graph with s strongly connected components, the number of strongly connected components in the new graph can equal any number between 1 and s, but cannot exceed s.

(g) $\sum_{i=1}^{n}(i^2 - i) = n(n+1)(2n+1)/6 - n(n+1)/2.$

(h) Run DFS on a directed graph G computing visit times *pre*(*v*) and *post*(*v*) for each vertex *v*. An edge (*u*,*v*) is a backedge if and only if *pre*(*v*) < *pre*(*u*) < *post*(*u*) < post(*v*).

(i) Adding one edge to a DAG must create a cycle.

(j) You filled in your name, your TAs name, and your key on the first page of this exam.