American University of Beirut

Advanced Algorithms & Data Structures

Spring 2003

CMPS 377

Final Exam

Date: Monday, June 9, 2003 - 3:00 pm to 5:00 pm
Instructor: Dr. Mohamed Kobeissi

Name: ....

ID #: ....

Section: **1**

This is NOT an open-book exam. However, you are allowed to have 2 papers format A4 on which you can write whatever you want. Your exam should have 10 pages, and there are 8 questions totaling 100 points. Your are NOT allowed to use any external notes. Your answers should be concise, and when possible should be a list of important points rather than prose. Solve as much problems as you can. I advise each one of you to pick the problems that he/she thinks easiest for him/her and work on them. I also advise you to spend time on understanding the problem and budget your time for solving each problem, or else you will wasting a lot of time on one problem and will run out of time for other problems.

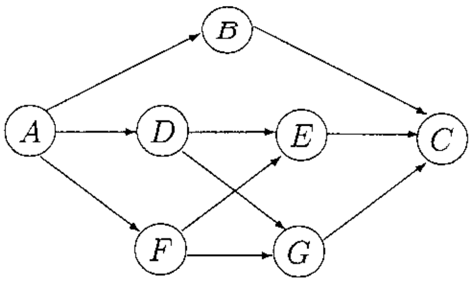
Wordy and/or irrelevant answers will reduce your score for that problem. Your answers should be the summary of work done on scratch paper that you do not hand in. If I could not read your writing, I will just give a ZERO without bothering myself trying to understand what you are writing. The space allocated for answers should be sufficient for your answers. If not, you can continue each exercise on the reverse side.

Good luck

Exercise 1 *(Multiple choice)* **[20 points]:**

Circle the correct answer. No explanation are required. Each correct answer is worth 2 points, but 1 point will be subtracted for each wrong answer, so answer only if you are reasonably certain.

1) Which of the following is *not* a possible Depth-First Search order for the graph below? *(by Depth-First Search order we mean the order in which Depth-First Search marks the vertices as having been visited.)*

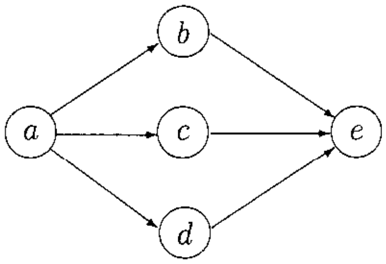


A. $ADECGBF$ B. $ABCDEGF$ C. $ABCDEFG$ D. $AFGCEBD$

2) Which of the following is *not* a possible Breadth-First Search order for the graph of the previous question? *(by Breadth-First Search order we mean the order in which Breadth-First Search marks the vertices as having been visited.)*

A. $AFBDEGC$ B. $ABDFCEG$ C. $ABDFEGC$ D. $ADBFECG$

3) How many topological sort exist for the following graph?



A. 3 B. 6 C. 9 D. 12 E. 15

4) What is the smallest possible height of a decision tree with 200 leaves?

A. 9 B. 8 C. 7 D. 6 E. 5

5) Which sorting algorithm below can be viewed as using the divide-and-conquer approach, with all comparisons of keys made in the dividing phase?

A. Insertion sort B. Mergesort C. Quicksort D. Heapsort

6) Which of the following choices below describes the amount of time used by the following code?

$$
\begin{aligned}
&\textbf{for}(i = 0; i < n; i++)\\
&\quad \textbf{for}(j = 1; j < n; j = 2 \times j)\\
&\qquad \textbf{for}(k = 1; k < n; k = 2 \times k)\\
&\qquad\quad x = x + 1;
\end{aligned}
$$

A. $O(n)$ B. $O(n \lg n)$ C. $O(n \lg^2 n)$ D. $O(n^2)$ E. $O(n^2 \lg n)$

7) The solution of $T(n) = 4T(n/2) + n \lg n$ is

A. $O(n^2)$ B. $O(\sqrt{n} \lg^2 n)$ C. $O(n \lg n)$ D. $O(\sqrt{n} \lg n)$ E. $O(n)$

8) For advising, prerequisites are to be observed. In providing automatic advising, graphs are to be uses to model the situation, where vertices represent courses, and a directed edge is drawn from course 1 to course 2 if course 1 is a prerequisite for course 2. Which of the following graph algorithms is appropriate?

A. DFS B. Dijkstra C. BFS D. Topological sort

9) What is the minimum number of elements in a heap of height 4?

A. 31 B. 16 C. 4! D. 6

10) The maximum number of edges that can be removed from a graph with n vertices in order to construct a spanning tree is

A. $n(n-1)/2 - (n-1)$ B. $n(n-1)/2$ C. $n-1$ D. $(n-1)/2$

Exercise 2 *(True or False)* **[20 points]:**

Circle the correct answer, no justification is required.

1) Given a set of n elements, one can output in sorted order the k elements following the median in sorted order in time $O(n + k \lg k)$.

True False

2) In a Depth-first search of an undirected graph, there might be forward and cross edges.

True False

3) There might be back edges in a breadth-first search of an undirected graph.

True False

4) Suppose that a graph G has two edge-disjoint spanning trees (two trees that have no edges in common), then in G, every pair of vertices forms part of a cycle.

True False

5) After we run DFS on a directed graph G, it is possible for a single vertex $v \in G$ to be simultaneously incident on at least one back edge, forward edge, cross edge and tree edge (i.e. all these types of edges together).

True False

6) The solution of $T(n) = T(n/5) + O(n)$ is $O(n)$.

True False

7) In a graph G, the sum of the degrees of all the nodes of G is an even number.

True False

8) If there is a path from u to v in a directed graph G, and if $d[u] < d[v]$ in a depth-first search of G, then v is a descendant of u in the depth-first forest produced.

True False

9) If the depth-first search of a graph G yields no back edges, then the graph G is acyclic.

True False

10) Let P be a shortest path from some vertex s to some other vertex t in a graph. If the weight of each edge in the graph in increased by one, P remains a shortest path from s to t.

True False

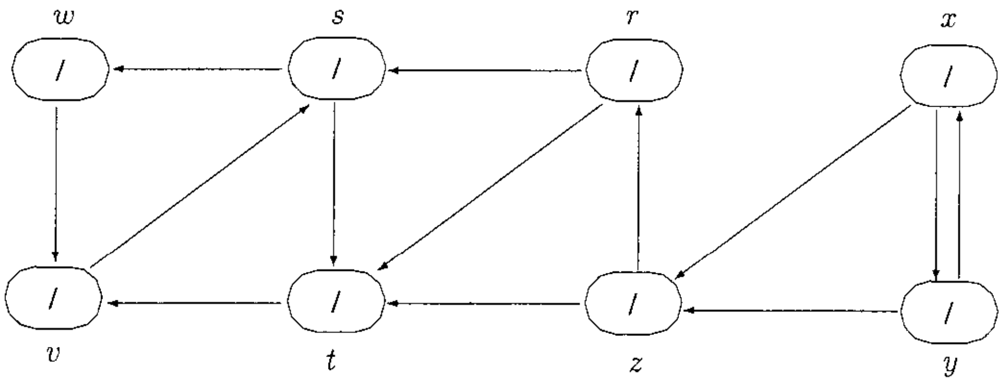
Exercise 3 *(BFS)* **[8 points]:**

a. Prove that in a breadth-first search of an undirected graph, for each tree edge (u, v), we have $d[v] = d[u] + 1$.

b. Prove that in a breadth-first search of an undirected graph, there are no forward edges.

Exercise 4 *(DFS)* **[12 points]:**

a. Apply depth-first search to the following graph, computing for each vertex its start and finish times. Label each edge as either (T) Tree, (F) Forward, (B) Back or (C) Cross. Apply DFS-visit to vertices in alphabetical order, starting from vertex r.



b. Show the depth-first forest produced.

Exercise 5 *(Minimum Spanning Tree)* [**8 points**]:

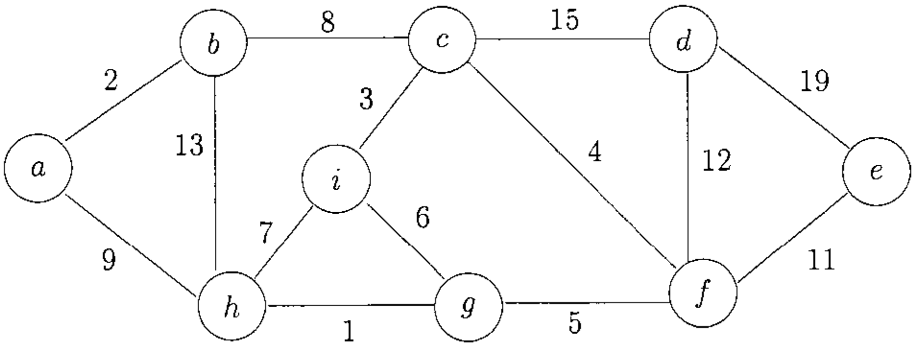
If all edges in a graph have different lengths, then there is only one Minimum Spanning Tree.

a. Prove that the MST must contain the shortest edge and the second-shortest edge (assume $|E| \geq 2$).

b. Give an example of such a graph (with at least 3 edges) in which the MST does not contain the 3rd-shortest edge.

Exercise 6 *(Prim's Algorithm)* **[8 points]:**

Consider the weighted graph G below.



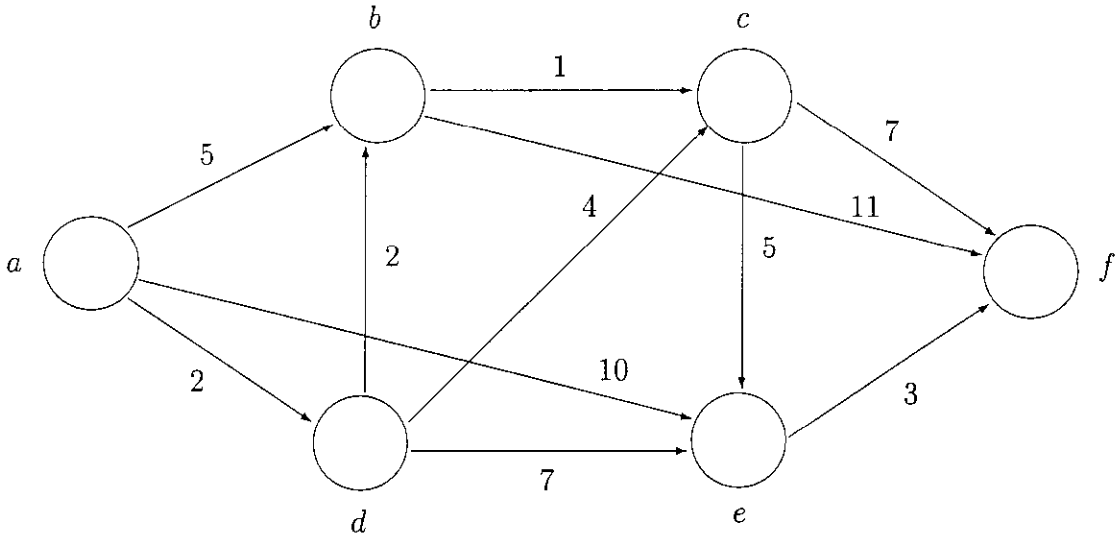
a. Give a light edge that crosses the cut $(\{a, b, c, f\}, \{d, e, g, h, i\})$.

b. Give a cut that respects the set of edges $\{(a, b), (a, h), (h, g), (c, i)\}$, and give a light edge that crosses this cut.

c. Show the MST as computed by Prim's algorithm, using vertex a as root. *(show only the final stage).*

Exercise 7 *(Dijkstra's Algorithm)* **[10 points]:**

a. Run Dijkstra's algorithm on the following graph, using vertex a as source. Fill out the d value for each vertex in the graph.



b. Draw the resulting shortest-paths tree.

Exercise 8 *(longest simple path)* **[14 points]**:

A simple path (of length l) in G is a path (of length l, that is, containing $l + 1$ vertices and l edges) with all vertices distinct.

Let $G = (V, E)$ be a directed acyclic graph represented as an adjacency list. Let s be a vertex in V, and let for every vertex $v \in V$ define $\mu(v)$ to be the length of the longest (in terms of the number of edges) simple path from s to v in G (if there is no path from s to v in G, then we define $\mu(v) = -\infty$).

a. Design an efficient algorithm that computes the values $\mu(v)$ for all vertices $v \in V$.

b. Provide some arguments behind the correctness of your algorithm.

c. Provide an analysis of the worst-case running time of your algorithm.