American University of Beirut

CMPS 256

Algorithms and Analysis

Fall 2002-2003

Final Exam

Date: Monday, January 21, 2003 - 11:30 am to 1:30 pm
Instructor: Mohamed Kobeissi

Name: ....

ID #: ....

Section: ....

This is an open-book, open-note exam. Your exam should have 11 pages, and there are 8 questions totaling 100 points. You may use your notes, class handouts, and the course textbook. 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. My guess is that the problems are going form the easiest to the hardest.

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, use additional papers.

Good luck

Exercise 1 [16 points]:

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

1. To determine of two binary search trees are identical trees, one could perform an inorder tree walk on both and compare the output lists.

ANSWER: TRUE FALSE

2. Your are given a graph in which all edges have non-negative weights, except those leaving the source vertex s. Then Dijkstra's algorithm may fail to correctly determine the shortest paths from s.

ANSWER: TRUE FALSE

3. Suppose T is a minimum spanning tree of a graph G. Then the largest edge weight of T is no larger than the largest edge weight of any other spanning tree of G.

ANSWER: TRUE FALSE

4. $n^{1+\varepsilon}$ is $O((1+\varepsilon)^n)$ for a fixed constant $0 < \varepsilon < 1$.

ANSWER: TRUE FALSE

5. Finding the minimum value in a max-heap has a $\Theta(n)$ complexity.

ANSWER: TRUE FALSE

6. By Case 2 of the Master Theorem, the solution to the recurrence $T(n) = 3T(n/3) + O(\lg n)$ is $T(n) = O(\lg n)$.

ANSWER: TRUE FALSE

7. When memory is limited, one should use **Heap-Sort** instead of **Merge-Sort**.

ANSWER: TRUE FALSE

8. A connected graph in which no two edges have the same weight has exactly one minimum spanning tree.

ANSWER: TRUE FALSE

Exercise 2 *(Asymptotics and Recurrences)* **[12 points]:**

The following is a Divide-and-Conquer algorithm for finding the minimum value in an (unsorted) array $S[1..n]$. The library function $\min(a, b)$ used in the algorithm returns the smaller between a and b.

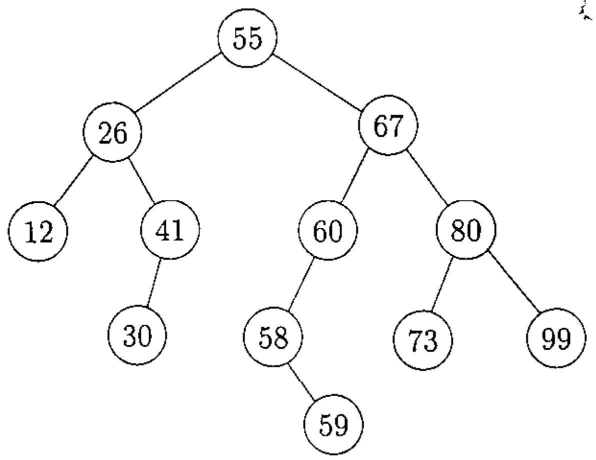
```
function minimum(S, x, y)
        if y − x ≤ 1
            then return min(S[x], S[y])
        else
            min₁ ← minimum(S, x, ⌊(x + y)⌋/2)
            min₂ ← minimum(S, ⌊(x + y)⌋/2 + 1, y)
        return min(min₁, min₂)
```

1. Write a recurrence relation for the worst-case number of comparisons used by minimum$(S, 1, n)$. *(you may assume that* n *is a power of* 2*).*

2. What is the running time of minimum$(S, 1, n)$. Justify your answer.

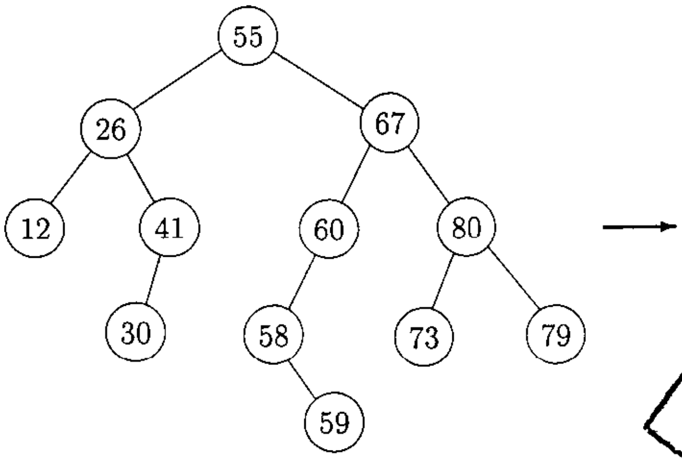
Exercise 3 *(Binary Search Tree)* **[6 points]:**

You are given a binary search tree as below:



1. What is the successor of 55? Mark out the path in the tree that finds this successor. What is the predecessor of 73? Mark out the path that finds this predecessor.

2. Draw the binary search tree resulting from the deletion of 55 from T.



Exercise 4 *(Stongly connected components)* **[10 points]:**

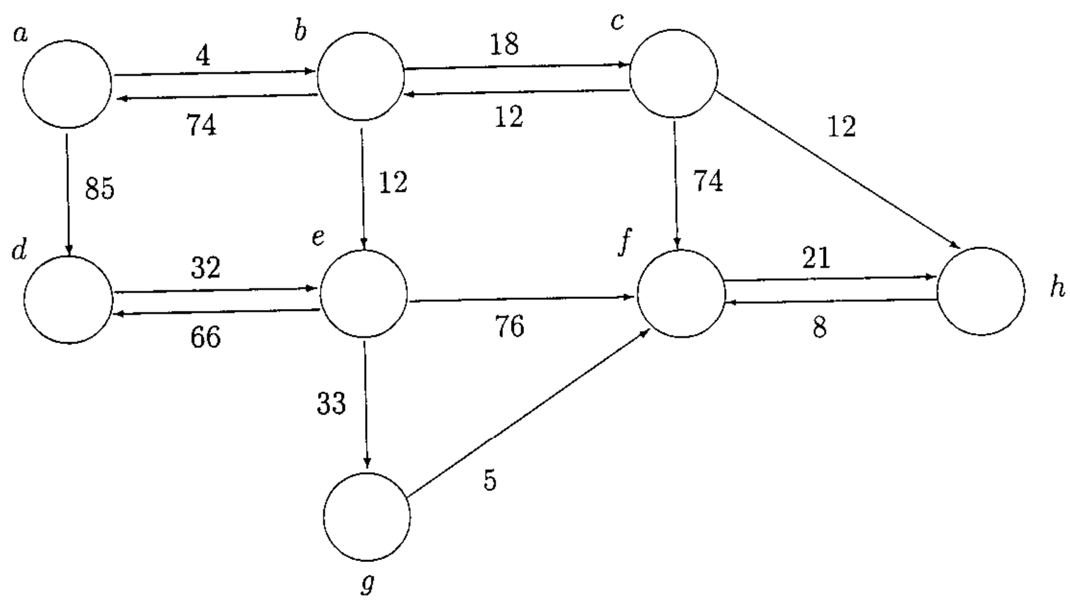
Consider a directed graph $G = (V, E)$. Let F be a DFS forest of G.

1. If vertices u and v are in the same strongly connected component of G, are they both in the same tree in F? argue why or give a counter example.

2. If vertices u and v are both in the same tree in F, are they both in the same strongly connected component of G? argue why or give a counter example.

Exercise 5 *(Dijkstra's algorithm)* **[13 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.

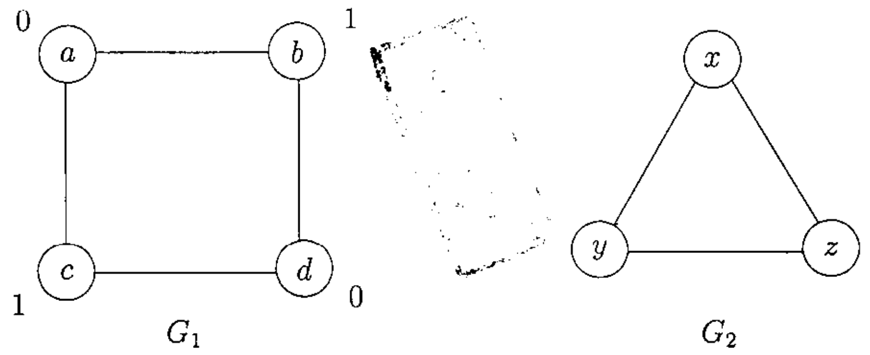


b. Draw the resulting shortest-paths tree.

Exercise 6 *(Breadth First Search)* **[15 points]:**

A *2-coloring* of an undirected graph G is a coloring of the vertices of G with two different colors, where adjacent vertices have different colors.

A graph is 2-*colorable* if a 2-coloring of the graph is possible. For example, graph G_1 below is 2-colorable (with each vertex "colored" 0 or 1) while graph G_2 is not.



1. Is a tree 2-colorable? Why or why not?

2. We first run the BFS algorithm on some vertex of a graph $G = (V, E)$. This constructs a BFS-tree T and defines $d(u)$ and $d(v)$, for all $v \in V$, where $d(v)$ is the depth of v in T.

   (a) For any non-tree edge (u, v), what is the relationship between $d(u)$ and $d(v)$?

   (b) For any non-tree edge (u, v), what would the relationship between $d(u)$ and $d(v)$ be if the graph was 2-colorable?

   (c) Now, give the algorithm for 2-colorability checking. Assume that you have a subroutine for the BFS algorithm and you have computed $d(u)$ for all $u \in V$.

Exercise 7 *(Checking the Shortest-Paths)* **[10 points]:**

Given an undirected graph $G = (V, E)$, a weight function $w :\to \mathbf{R}$, and a subset of the graph vertices $U \subset V$. Discuss which of the two following algorithms for MST (Prim's and Kruskal's) would you choose if a connected MST is required that spans the vertices of U. Discuss also the implication on the worst case running time.

Exercise 8 [18 points]:

A directed graph $G = (V, E)$ is *unipathic* if for any two vertices $u, v \in V$, there is at most one simple path from u to v. Suppose a unipathic graph G has both positive and negative edge weights.

1. Design an $\Theta(V)$-time algorithm to determine the shortest-path weights from a source s to all vertices $v \in V$ for such a unipathic graph. If some shortest-path weights to vertices reachable from s do not exist, your algorithm should report that a negative-weight cycle exists in the graph.
2. Provide some arguments behind the correctness of your algorithm.
3. Justify the running time of your algorithm.