Name: ______________________________

Algorithms
Fall 2018
Midterm

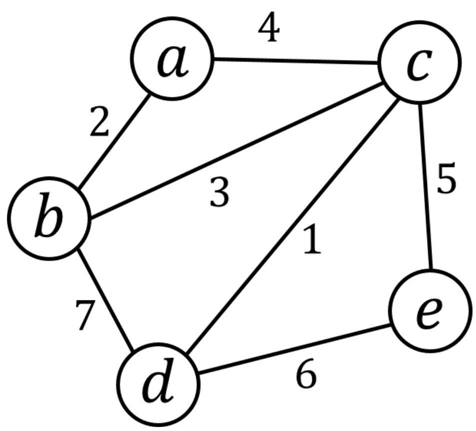
This is a 50-minute, closed-notes exam. **Please put away all notes, phones, laptops, etc.** There are eight pages and four problems. Good luck!

Problem	**Score**
1	/26
2	/27
3	/27
4	/20
Total	/100

You do not need to show any work beyond your final answer, but you are welcome to do so if you feel it will help me understand your thought process.

1 Minimum spanning trees.

Here's a graph:

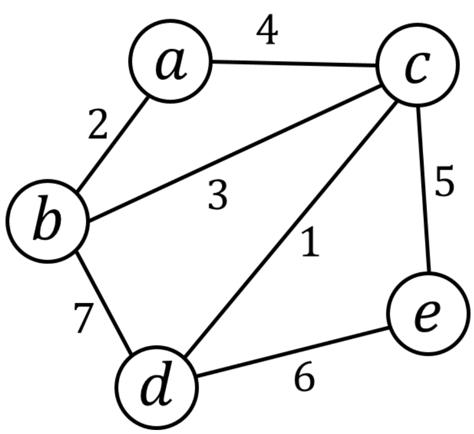


(a) Run Prim's algorithm, starting from node a, to find a minimum spanning tree for this graph. In what order are edges added to the MST?

(b) Jamie suggests the following alternative algorithm for finding a minimum spanning tree:

```
JamieMST(G=(V,E))
    T = edges in T, sorted in decreasing order of weight
    for each edge (u,v) in T
        if edge (u,v) is part of some cycle in T
            remove edge (u,v) from T
    return T
```

Run Jamie's algorithm on the same graph (reproduced below). In what order are edges deleted from the graph, and what is the result?



(c) Under what circumstances is it "safe" to add an edge to a partial MST? Suppose edge (u, v) was deleted by Jamie's algorithm. Could edge (u, v) be a "safe" edge for any partial MST (assuming all edge weights in the graph are distinct)?

2 Knapsacks.

In the Fractional Knapsack Problem, we had a set of n items, where each item i has value v_i and weight w_i. We also had a knapsack that could hold up to total weight W. Our goal was to determine what fraction p_i to take of each item i in order to maximize the value stored in the knapsack without exceeding the weight limit.

(a) What was the greedy choice to decide what item to steal first that led to an optimal solution?

(b) Now suppose that we must set $p_i = 0$ or $p_i = 1$ for each item; that is, we must take all of the item or none of it. This is called the 0-1 Knapsack Problem. Give a counterexample that shows that the greedy choice you made in part (a) no longer is guaranteed to find the optimal solution.

(c) Let $V(i, W)$ denote the optimal value achievable for the 0-1 Knapsack Problem given i items and a weight limit of W. Write a recurrence that expresses $V(i, W)$ in terms of solutions to smaller problems.

3 Which is better?

Mackenzie and Asa are trying to solve a new computational problem. Each of them proposes a divide-and-conquer algorithm.

- Mackenzie's algorithm takes time $5n^2$ to divide the problem into four pieces, each of size $n/4$, then solves the pieces recursively.
- Asa's algorithm splits the problem into sixteen pieces, each of size $n/4$, solves the pieces recursively, then takes time $3n$ to combine the results of the pieces.

Mackenzie claims that her algorithm is faster because she creates fewer subproblems. Asa argues that his algorithm is faster because he does less work in the "divide" and "combine" steps. Your job is to figure out who is right.

(a) Write a recurrence for $T_M(n)$, the runtime of Mackenzie's algorithm, and write a recurrence for $T_A(n)$, the runtime of Asa's algorithm.

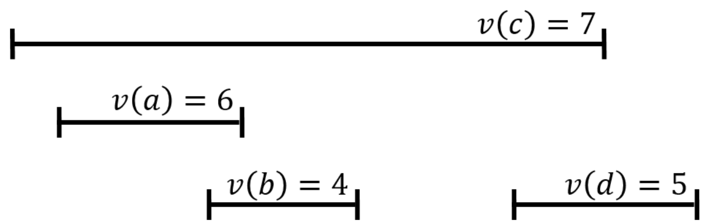
(b) Use the Master Theorem to solve each of the recurrences you wrote in part (a).

(c) Based on your answer in part (b), who is right? Is Mackenzie's algorithm or Asa's algorithm faster?

(There's a lot of empty space here but absolutely no need to write a lot of words.)

4 Weighted interval scheduling.

Here's a set of intervals with associated weights:



(a) Run the dynamic programming algorithm discussed in class to find the maximum value of any set of non-overlapping intervals. In your solution, it should be clear both what value the algorithm returns and how it goes about finding the optimal solution.

(b) Suppose that you want to find the set of intervals that is included in the optimal solution, rather than just the value of that solution. How would you go about reconstructing the optimal set of intervals?