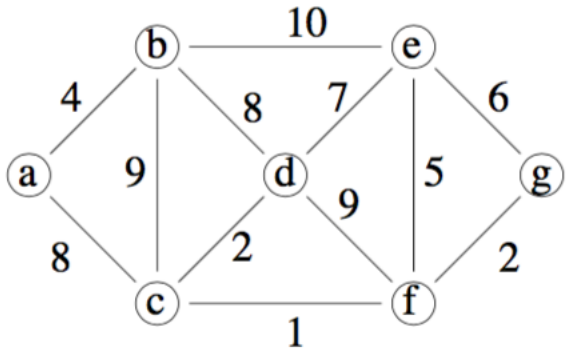
Homework 6
Com S 311
Due: Dec 5, 11:59PM
(Late Submissions are NOT accepted)

1. Consider the following graph:



   (a) Use Prim's algorithm to find the minimum spanning tree.

   (b) Use Dijkstra's algorithm to find the shortest path from e.

   (20pts)

2. A new algorithm for minimum spanning tree is being developed. The steps of the algorithm are as follows.

```
Given an undirected graph containing V vertices, E edges
(with +ve edge weights).
Mark all edges as not-visited
Initialize MSTree to include all edges in the graph.

Repeat
 pick the not-visited edge with the highest weight from MSTree
 if the removal of the edge does not make disconnected components
    then remove the edge from the MSTree
 else mark the edge as visited
until no edges can be removed

return MSTree as the MST of the graph.
```

   Prove or disprove the correctness of the above algorithm. To prove correctness, you will need to argue that the returned MSTree is indeed a MST of the input graph. To disprove correctness, you can present a counter-example, i.e., an input graph for which the above algorithm returns MSTree which does not represent its MST. (40pts)

3. In a directed graph, if a vertex can reach all other vertices, then we will call it a dominating vertex. Design an algorithm to find the dominating vertices in a graph. Prove its correctness and explain the time complexity of your algorithm. (40pts)

4. Let A be a set of integers $\{a_1, \ldots, a_n\}$. Write an algorithm that gets A as input and determines if A has a subset A_1 such that

$$\sum_{x \in A_1} x = \frac{a_1 + a_2 + \ldots + a_n}{2}$$

I.e, sum of elements in A_1 is exactly half of $(a_1 + \ldots + +a_n)$. Your algorithm must use dynamic programming paradigm. First write the recurrence relation, explain the correctness of the recurrence relation. Design an algorithm based on the recurrence relation. State and derive the time bound of the algorithm. Your algorithm should not use recursion. (50pts)

5. You are in a rectangular maze organized in the form of $M \times N$ cells/locations. You are starting at the upper left corner (grid location: $(1, 1)$) and you want to go to the lower right corner (grid location: (M, N)). From any location, you can move either to the right or to the bottom, or go diagonal. I.e., from (i, j) you can move to $(i, j+1)$ or $(i+1, j)$ or to $(i+1, j+1)$. Cost of moving right or down is 2, while the cost of moving diagonally is 3. The grid has several cells that contain diamonds of whose value lies between 1 and 10. I.e, if you land in such cells you earn an amount that is equal to the value of the diamond in the cell. Your objective is to go from the start corner to the destination corner by picking up diamonds such that the difference between the sum of the value of diamonds you picked and sum of the costs incurred. maximized.

   (a) Write a greedy algorithm to address the problem. Show that the algorithm does not always produce the path that will give you the maximum number of diamonds.

   (b) Write a dynamic programming algorithm to address the problem. Analyze its time complexity.

(50 pts)

GUIDE LINES:

- It is important to know whether your really know! For each problem, if you write the state- ment I do not know how to solve this problem (and nothing else), you will receive 20% credit for that problem. If you do write a solution, then your grade could be anywhere be- tween 0% to 100%. To receive this 20% credit, you must explicitly state that you do not know how to solve the problem.

- You must work on the homework problems on your own. You should write the final solutions alone, without consulting any one. Your writing should demonstrate that you understand the proofs completely.

- When proofs are required, you should make them both clear and rigorous. Do not hand waive.
- If you hand writing is not legible, then your homework will not be graded.
- Any concerns about grading should be made within one week of returning the homework.
- Please submit your HW via blackboard. If you type your solutions, then please submit pdf version. If you hand-write your solutions, then please scan your solu- tions and submit a pdf version. Please make sure that the quality of the scan is good, and your hand writing is legible. Name your file must be YourNetID-HW6.pdf. For example, if your net id is bondj, then the file name should be bondj-HW6.pdf. HWs submitted in incorrect format (non pdf, incorrect file name etc) will incur a penalty of 30%.