Test 2

First Name: _____

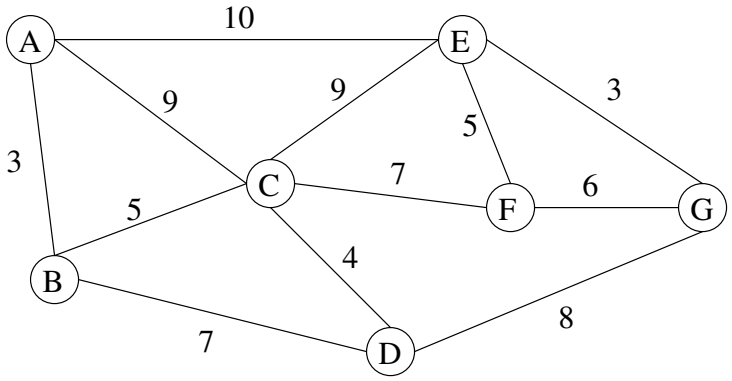
Last Name: _____

Student Number: _____

- *This test lasts* 75 minutes.
- *Aids allowed: handwritten notes on both sides of one* 8.5×11 *inch page.*
- *Make sure your test has 6 pages, including this cover page.*
- *Answer in the space provided. (If you need more space, use page 6 and indicate clearly that your answer is continued there.)*
- *You may use any algorithm that was covered in class or the readings without explaining how it works. You may use any result that was proved in class or the readings without reproving it.*

Question 1	/3
Question 2	/6
Question 3	/3
Question 4	/5
Question 5	/6
Total	/23

[3] **1.** Draw the minimum spanning tree that would be found by Kruskal's algorithm for the graph shown below.



[6] **2.** Let $A[1..n]$ be an array of numbers. You must implement an operation RangeQuery(a, b) that returns the *number* of elements in A that are greater than a and less than b. More formally, RangeQuery(a, b) should return $|\{i : 1 \leq i \leq n \text{ and } a < A[i] < b\}|$. Your goal is to minimize the total time to perform k RangeQuery operations. (The array A is the same for all k queries.)

(a) Briefly describe an efficient solution if $k = 5$. Use Θ notation to describe the worst-case total running time for a sequence of 5 RangeQuery operations.

(b) Briefly describe an efficient solution if $k = \lfloor\sqrt{n}\rfloor$. Use Θ notation to describe the worst-case total running time for a sequence of $\lfloor\sqrt{n}\rfloor$ RangeQuery operations.

[3] **3.** Suppose each element of array $A[1..n]$ is from the set $\{1, 2, \ldots, n^3\}$. Explain briefly how you could sort the array in $O(n)$ worst-case time.

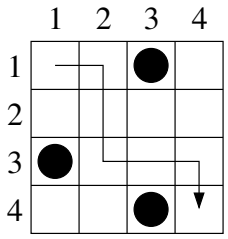
[5] **4.** Suppose you are given n jobs $J_1, \ldots, J_n$. Job J_k will require d_k days to complete. Your goal is to pick a the largest possible subset of the jobs such that the total time to complete all jobs is at most D days. Fill in the box below to create a correct greedy algorithm for this problem.

Sort the jobs so that [ ]
$S = \emptyset$ % set of jobs to perform
$x = D$ % number of available days
for $k = 1..n$
  if $d_k < x$ then
    $S = S \cup \{J_k\}$
    $x = x - d_k$
  end if
end for
output S

Let S_i be the value of S after i iterations of the loop. Give a formal definition of what it means for an optimal solution S^* to be an extension of S_i.

Prove the following: If there is an optimal solution S^* that extends S_{i-1} and $J_i \in S_i$, then there is an optimal solution $\hat{S}$ that extends S_i.

[6] **5.** A region is divided into n^2 squares, some of which are blocked by obstacles. (In the example shown, $n = 4$ and there are three obstacles). You are given a 2-dimensional array $A[1..n, 1..n]$. If there is an obstacle in row i and column j, then $A[i, j] = 0$; otherwise $A[i, j] = 1$. A path consists of a sequence of steps from one square to an adjacent square. We are interested in paths that travel from the northwest corner to the southeast corner such that every step along the path moves either east or south and does not pass through any square that contains an obstacle. Design a dynamic programming algorithm that would efficiently compute the *number* of such paths.



(a) Define the array that will hold the solutions to subproblems. Explain in English what the value stored in each location of this array will represent.

(b) Give the equation(s) you would use to compute each entry of the array.

(c) Briefly describe the order in which you would fill in entries of the array.

(d) Use Θ notation to give the worst-case running time of your algorithm in terms of n.

This nearly blank page is just for additional workspace, if you need it.