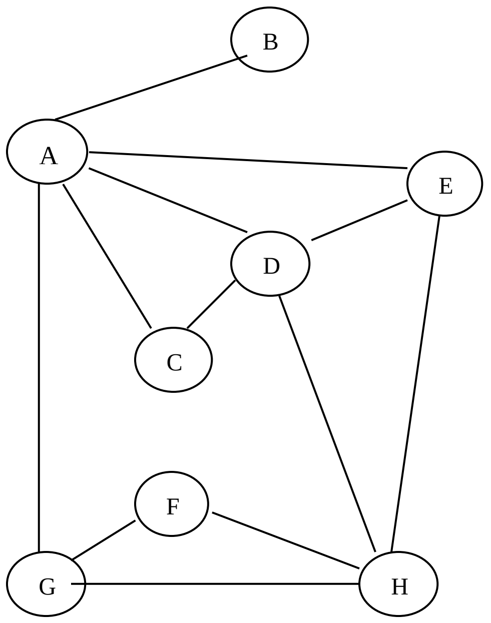
CS372 Assignment #7.

Due: at the beginning of the lecture on Thursday, April 25.

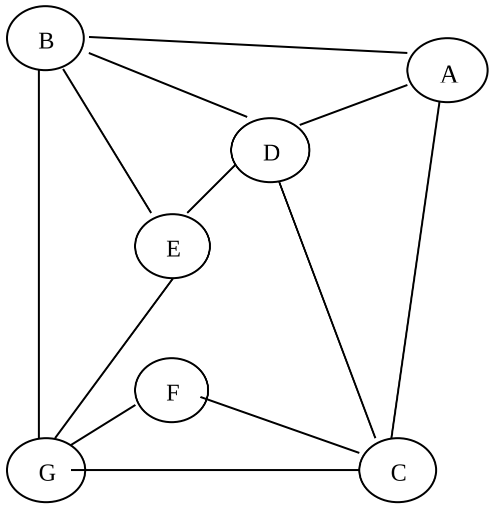
1. Consider the following graph with edge weights listed in the table.



AB	5
AC	4
AD	4
AE	6
AG	8
CD	1
DE	2
DH	7
EH	7
FG	1
FH	3
GH	6

(a) What is the cost of its minimum spanning tree?
(b) How many minimum spanning trees does it have?
(c) Suppose Kruskal's algorithm is run on this graph. In what order are the edges added to the MST (in case of ties add the edge which is lexicographically first, that is, comes first in the table)? For each edge in this sequence, give a cut that justifies its addition.

2. Suppose we want to find the minimum spanning tree of the following graph.



AB	5
AC	4
AD	2
BD	6
BE	3
BG	1
CD	6
CF	7
CG	5
DE	1
EG	3
FG	1

(a) Run Kruskal's algorithm (in case of ties add the edge which is lexicographically first, that is, comes first in the table). Show how the disjoint-sets data structure looks at every intermediate stage (including the structure of the directed trees), assuming path compression is **not** used.

(b) Run Prim's algorithm; whenever there is a choice of nodes, always use alphabetic ordering (e.g., start from node A). Draw a table showing the intermediate values of the `cost/prev` values.

3. The following table gives the frequencies of the symbols in a particular document.

a 10%	c 20%	e 12%	g 15%	i 3%
b 7%	d 5%	f 6%	h 14%	j 8%

(a) What is the optimum Huffman encoding of this alphabet?
(b) If this encoding is applied to a file consisting of 1,000,000 characters with the given frequencies, what is the length of the encoded file in bits?

4. We use Huffman's algorithm to obtain an encoding of alphabet {a, b, c, d} with frequencies f_a, f_b, f_c, f_d.
Assume that $f_a \geq f_b \geq f_c \geq f_d$. List all possible different coding trees that could be obtained (consider two trees to be the same if one tree can be obtained from the other one by changing the order of children of nodes and/or exchanging 0/1 labels on edges connecting nodes to their children). For each coding tree give an example of frequencies (f_a, f_b, f_c, f_d) that would yield the coding tree.

5. Consider the following problem. You are organizing a conference. Let T ={t_1,...,t_n} be a list of times when participants arrive at the airport. When a participant arrives at the airport there must be a student volunteer at the airport to meet the participant and direct him/her to a shuttle. Each student volunteer should spend no more than 1 hour at the airport. You need to come up with a schedule that uses the smallest number of student volunteers and has all the times in T covered.
(a) Does the following algorithm correctly solve the problem? Let I be a one hour time interval that covers the most number of points in T. Add I to the solution set S. Then recursively continue on the points in T not covered by I.
If the algorithm is not correct, then provide an example set T on which it produces a wrong result.
(b) Does the following algorithm correctly solve the problem? Let tj be the smallest (leftmost) point in T. Add the interval I = (t_j ; t_j + 1) to the solution set S. (The interval I is one hour long. It starts at time t_j and ends one hour later.) Then, recursively continue on the points in T not covered by I. If the algorithm is not correct, then provide an example set T on which it produces a wrong result.

6. Assume that there are n kinds of available coins. The denominations are 1 = a1 < a2 < · · · < an dollars, respectively, and each ai , $1 \leq i \leq n$ is an integer. Your goal is to make change for m dollars using the fewest number of coins. Assume that you are using the following greedy approach: always choose the biggest denomination that is smaller than the remaining amount. For example, if m=68 and the coin denominations are 25, 10, 5, and 1, then you would first choose two coins of 25, then one coin of 10, then one coin of 5, then three coins of 1 (68 = 2*25 + 1*10 + 1*5 + 3*1). The greedy approach in this example gave a solution with 7 coins. Give a set of denominations for which the greedy algorithm does not yield an optimal solution (solution with the fewest number of coins) for some m. Explain your answer by giving an example of m with an optimal solution and solution obtain by the greedy approach.