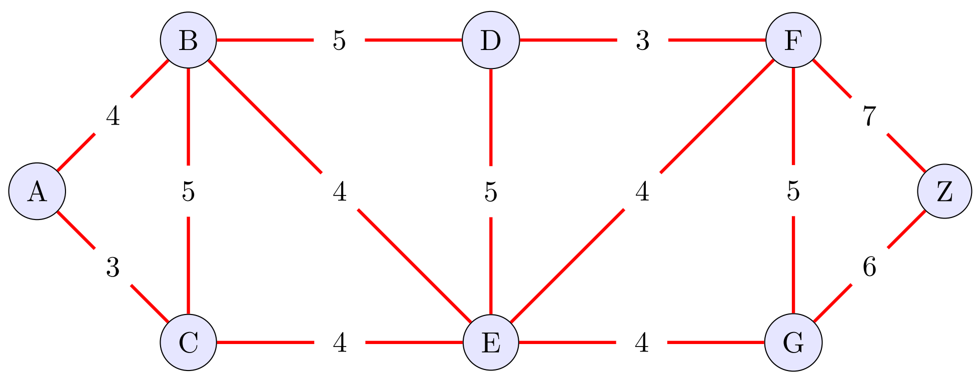
Data Sheet

do not submit

Graph K



Union by Rank

The current state S of the `union by rank` representation of disjoint subsets of the set of vertices {A, B, C, D, E, F, G, H, I} is given by

`pi`	**A**	**B**	**D**	**D**	**B**	**F**	**D**	**A**	**F**
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**

`rank`	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**

CIS 3223 HW 7

Dr Anthony Hughes

Name: Solutions
Temple ID (last 4 digits:

1 (9 pts) For the graph K use Kruskal's algorithm to find a **minimum**-cost spanning tree, and then determine the minimum cost. Use alphabetical ordering.

Construct a hash table using the lengths of the edges to store the edges.

Sort the edges in each bucket using alphabetical ordering.

Edge List (Hash Table)

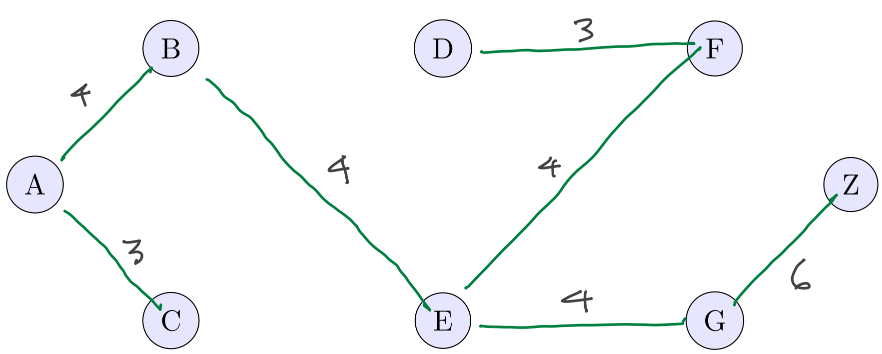
3	AC	DF			
4	AB	BE	EF	CE	EG
5	BD	BC	DE	FG	
6	GZ				
7	FZ				

Sorted Edge List (Hash Table)

3	AC	DF			
4	AB	BE	~~CE~~	EF	EG
5	~~BC~~	~~BD~~	~~DE~~	~~FG~~	
6	GZ				
7	FZ				

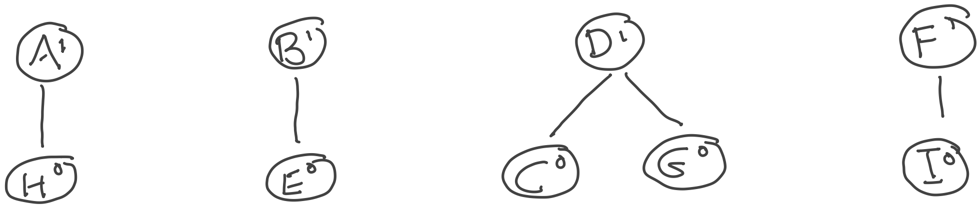
6
16
6
28

Construct a minimum cost spanning tree by using the edges in each bucket moving from left to right starting with bucket with the lowest value:



Minimum Cost 28

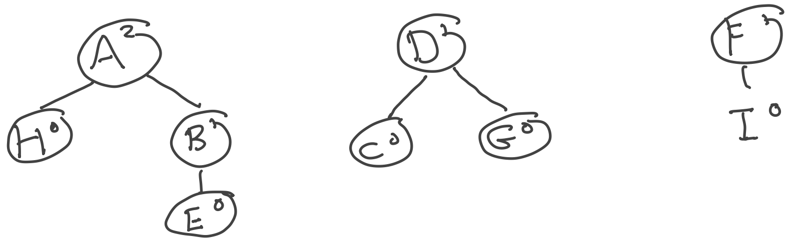
2 (11 pts) (a) For S, draw the corresponding trees representing the sets.



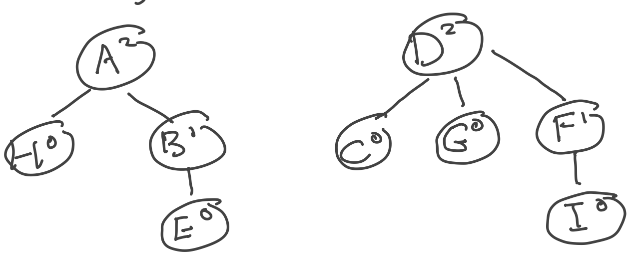
(b) Consider the following **SEQUENCE** of operations. Draw the corresponding tree(s) representing the sets after each of the operations has been executed (use alphabetical order):

(i) `union(A, E)` (ii) `union(F, G)` (iii) `union(C, I)` (iv) `union(E, F)`

`union(A, E)` find(A) = A, find(E) = B, ran(A) = 1, rank(B) = 1 join B to A
rank(A) = 2

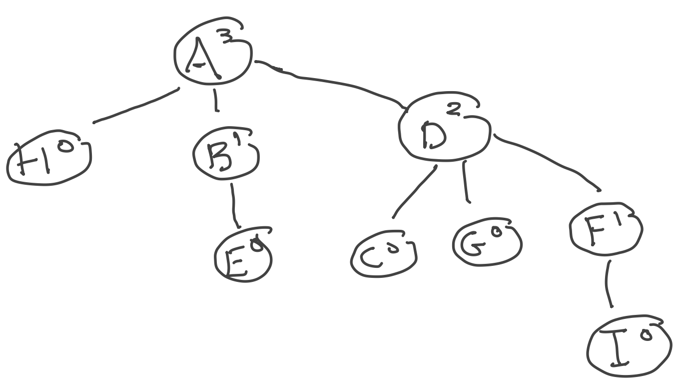


union(F, G) Find(F) = F, find(G) = D, rank(F) = 1, rank(D) = 1 join F to D
rank(D) = 2



union(C, I) find(c) = D, find(I) = D cycle

union(E, F) find(E) = A, find(F) = D rank(A) = 2, rank(D) = 2 join D to A
rank(A) = 3



(c) Specify the current state of `pi` and `rank` after the sequence has been executed.

pi

A	B	C	D	E	F	G	H	I
A	A	D	A	B	D	D	A	F

rank

A	B	C	D	E	F	G	H	I
3	2	0	2	0	1	0	0	0

(d) Use induction to show that if a subtree T constructed in the rank by union procedure has rank m, then T contains at least 2^m nodes.

Verify this result using the tree(s) drawn in (b)(iv).

root	A	B	C	D	E	F	G	H	I
rank	3	2	0	2	0	1	0	0	0
2^{rank}	8	4	1	4	1	2	1	1	1
|nodes|	9	2	1	5	1	2	1	1	1
Verified	T	T	T	T	T	T	T	T	T

Base case: $m = 0$:

T has one node (root).

$2^k = 2^0 = 1$

So true in this case.

Inductive case: Assume true for for $m = k$. Show true for $m = k + 1$

Let x be a node with $rank(x) = k+1$

x promoted to rank $k+1$ by joining y with rank k to x with rank k.

Before the union, the subtree with node x as root and the subtree with node y as root both have at least 2^k nodes.

But then after the union, the subtree with node x has at least $2^k + 2^k = 2^{k+1}$ nodes.

So true for $m = k+1$.