Yunus Emre TAT 25187
Question 1 (20 points)

Prove that the **average** running time of *quickselect* algorithm (given in Figure 7.16 in the second edition of your textbook) is $O(N)$. For simplicity, assume that 1) pivots are chosen randomly, 2) partition sizes are equally likely, and 3) no cutoff size is used. Note that this proof is quite similar to that of average running time of Quicksort algorithm.

Quickselect uses the same overall approach as quicksort,choosing one element as a pivot and partitioning the data two based on the pivot,accordingly as less than or greater than pivot.The difference is instead of reccuring for both sides,it recurs only for the part that contains the kth smallest element.So quickselect algorithm has complexity at average case O(N).
In this quesiton selection of the pivot element is random.
So,
K is the number of the left hand side numbers,N is the number of the array numbers,k/N is the probability of the located in the left hand side.

N-K-1 is the number of the right hand side numbers,(N-K-1)/N is the probability of the located in the right hand side.

1/N is the probability of the becoming pivot.

$$Quick\ sort\ algorithm: T(N) = T(i) + T(N - i - 1) + cN$$

$$\mathrm{T(N)} = \sum_{k=0}^{N-1} \frac{k}{N}\mathrm{T(k)} + \sum_{k=0}^{N-1} \frac{N-k-1}{N} T(N-k-1) + N$$

We used the basic quick sort relation to find the below relation because partions size equally likely,pivot chosed rondomly.Mean of the quick sort algorithm used to find average complexity.

$$T(N) = N + \frac{1}{N}\sum_{k=0}^{N-1}(\frac{k}{N}T(k) + \frac{N-k-1}{N}T(N-k-1) + \frac{1}{N}0)$$

All pivots are chosed randomly thats mean partition sizes are equally likely.It's mean :

$$\frac{1}{N}\sum_{k=0}^{N-1} \frac{k}{N}\mathrm{T(k)} = \frac{1}{N}\sum_{k=0}^{N-1} \frac{N-k-1}{N}\mathrm{T(N-k-1)}$$

So that our equation can be write such that

$$\mathrm{T(N)} = \frac{1}{N}\sum_{k=0}^{N-1}\frac{k}{N}\mathrm{T(k)} + \frac{1}{N}\sum_{k=0}^{N-1}\frac{k}{N}\mathrm{T(k)} + N$$

When multiply both sides with N^2

$$N^2\mathrm{T(N)} = N^3 + 2\sum_{k=0}^{N-1}\mathrm{kT(k)}$$

When I write the N-1 instead of N I will have the equation like that,

$$(N-1)^2\mathrm{T(N-1)} = (N-1)^3 + 2\sum_{k=0}^{N-2}\mathrm{kT(k)}$$

Now we want to get rid of the summation sign.For that,we have to substract last 2 equations above.

Finally I have

$$N^2\mathrm{T(N)} = 3N^2 - 3N - 1 + T(N-1)(N^2-1)$$

Now divide both sides by N^2

$$\mathrm{T(N)} = 3 - \frac{3}{N} - \frac{1}{N^2} + T(N-1)\frac{N^2-1}{N^2}$$

After the rearragnment we have equation ,

$$\mathrm{T(N)=T(N\text{-}1)+C}\text{ ,c is constant}$$

Now it is a recursion equation,

$$\mathrm{T(N)=1+cN}\text{ , }\mathrm{T(0)=1}$$

Finally,average case of quick select is O(N) proven.

Question 2 (20 points)

Trace the operation of Dijkstra's *weighted* shortest path algorithm for the graph given in Figure 1. Use vertex *E* as your start vertex.

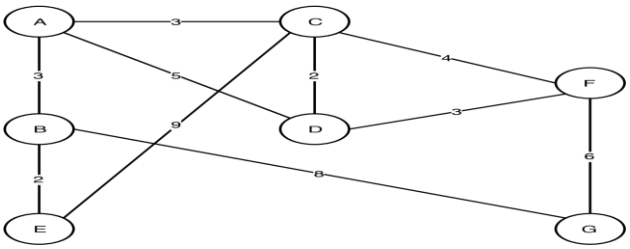
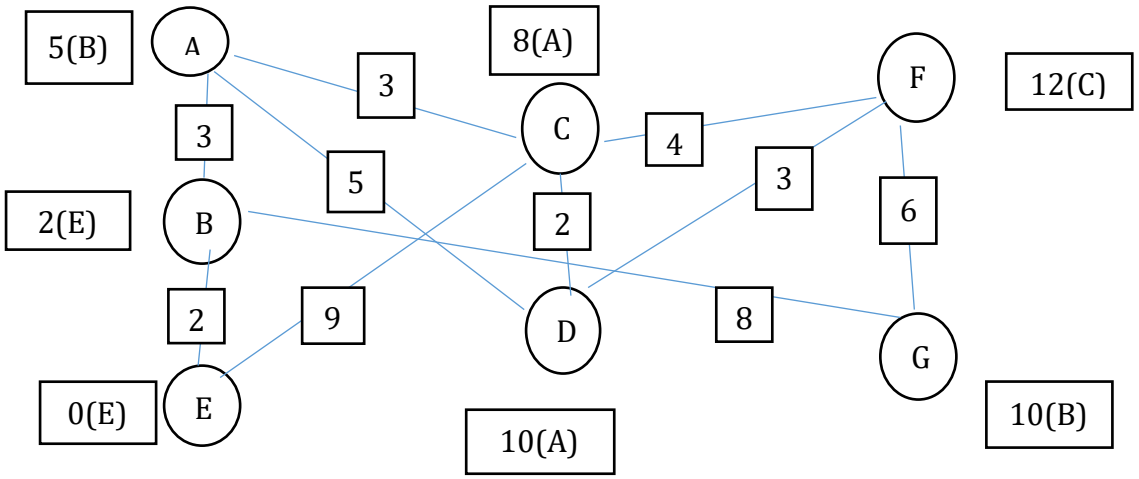


Figure1

X(Y): x is distance ,y is previous node

Initially all vertices' distances are equal to infinity, unknown and paths are empty.

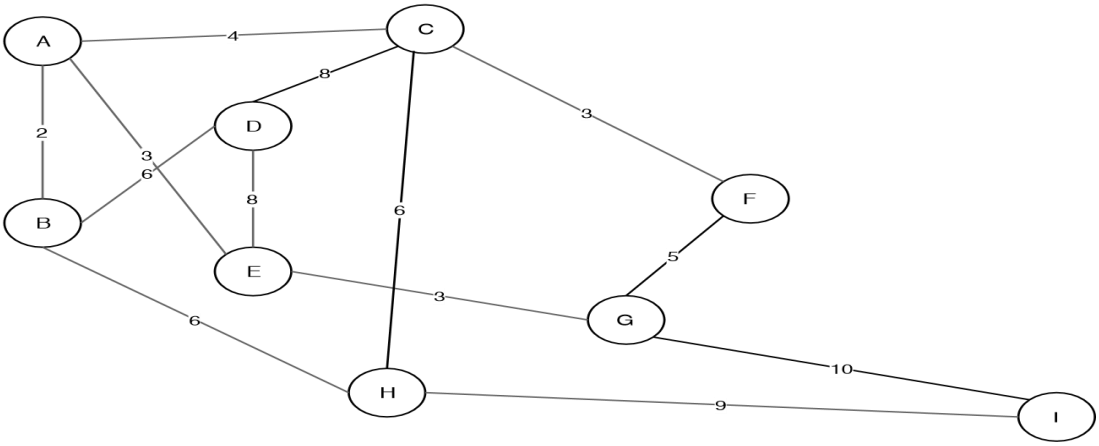
E marked as known, E recorded as 0(E).Look E's adjacent vertices B and C. B recorded as 2(E) and C recorded as 9(E) . Look all distances of unknown vertices and choose smallest distance which is B so mark B as known. Look B's adjacent vertices A and G. A recorded as 5(B) and G recorded as 10(B). Look all distances of unknown vertices and choose smallest distance which is A so mark A as known. Look A's adjacent vertices C and D. C recorded as 8(A) (because now C distance shorter than last record) and D recorded as 10(A). Look all distances of unknown vertices and choose smallest which is C so mark C as known. Look C's adjacent vertices F and D . F recorded as 12(C) and D is not recorded(10(C) is not shorter than last record.). Look all distances of unknown vertices and choose smallest which is D and G, pick D so mark D as known. Look adjacent vertex of F but do not change it because its path from C is 12 is smaller than path from D 13. Look all distances of unknown vertices and choose smallest which is G so mark G as known. Look adjacent vertex of G but do not change it because F path from C 12 is smaller than path from G 16. Look all distances of unknown vertices and choose smallest which is F so mark it as known.Look the F's adjacent vertices D and F.F from D distance is 15.It is not smaller than 10(A) D's previous and F from G is 18 it isnot smaller 10(B) don't change both.There is no unknown vertex yet so finished.



Question 3 (15 points)

Trace the operation of Prim's minimum spanning tree algorithm for the graph in Figure 2.

Use vertex *E* as your start vertex.



Starting with the vertex E marked as known.
1.)There are 3 edges that (3,8,3) connected vertex E.
I need to find the shortest edge connected to E.
E to G and E to A same.I selected to G ,marked as known added to tree .

2-)There are 4 edges that (3,8,5,10) connected to my known vertices.
I need to find the shortest edge connected to my known vertices.
E to A is 3.I selected to A,marked as known added to tree.

3-)There are 5 edges that(2,4,8,5,10) connected to my known vertices.
I need to find the shortest edge connected to my known vertices.
A to B cost is 2 (smallest).I selected to B,marked as known added to tree.

4-)There are 7 edges that (4,5,6,6,8,8,10) connected to my known vertices.
I need to find the shortest edge connected to my known vertices.
A to C cost is 4(smallest).I selected to C,marked as known added to tree.

5-)There are 7 edges that (3,5,6,6,8,8,10) connected to my known vertices.
I need to find the shortest edge connected to my known vertices.
C to F is 3(smallest) I selected to F,marked as known added to tree.

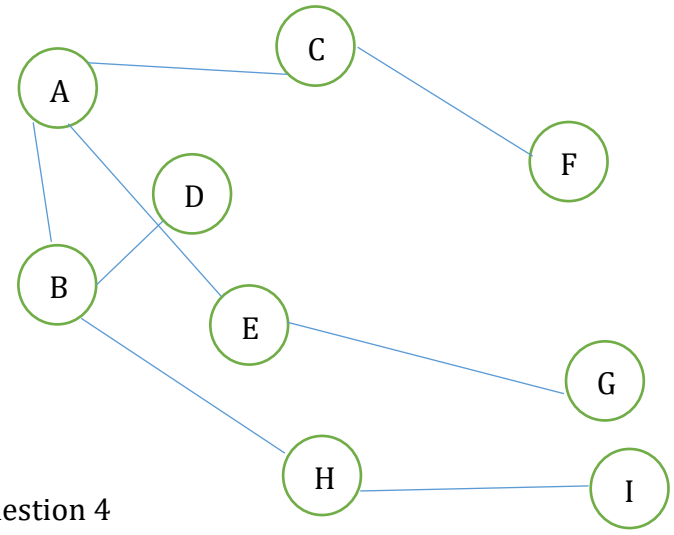
6-)There are 6 edges that (5,6,6,8,8,10) connected to my known vertices but 5 is F-G it is a cycle we can not select F-G .
I need to find the shortest edge connected to my known vertices.
B to D is 6(one of the smallest) I selected to D,marked as known added to tree.

7-) There are 4 edges that (6,8,8,10) connected to my known vertices.
I need to find the shortest edge connected to my known vertices.
B to H is 6(smallest) I selected to H,marked as known added to tree.
8-)There are 3 edges that (6,9,10) connected to my known vertices but C-H it is a cycle we cannot select.
I need to find the shortest edge connected to my known vertices.

H to I is 9(smallest) I selected to I,marked as known ,added to tree.

9-)There are 1 edges that (10) connected to my known vertices but G to I creates a cycle .

Algorithm is done



Question 4

Kruskal's algorithm

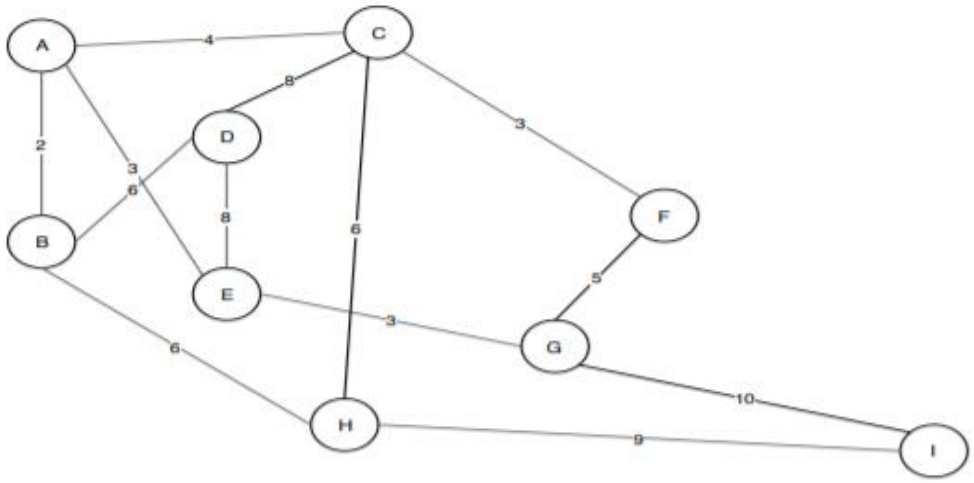
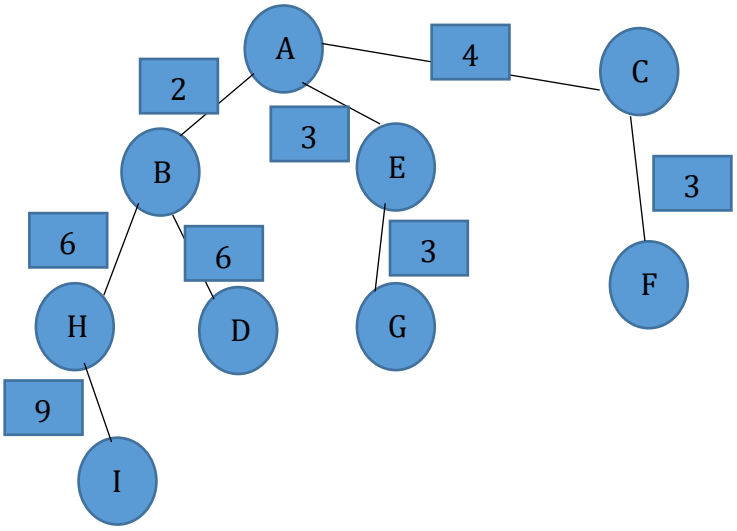


Figure 2: An undirected weighted graph.

Select the minimum edge without cycle

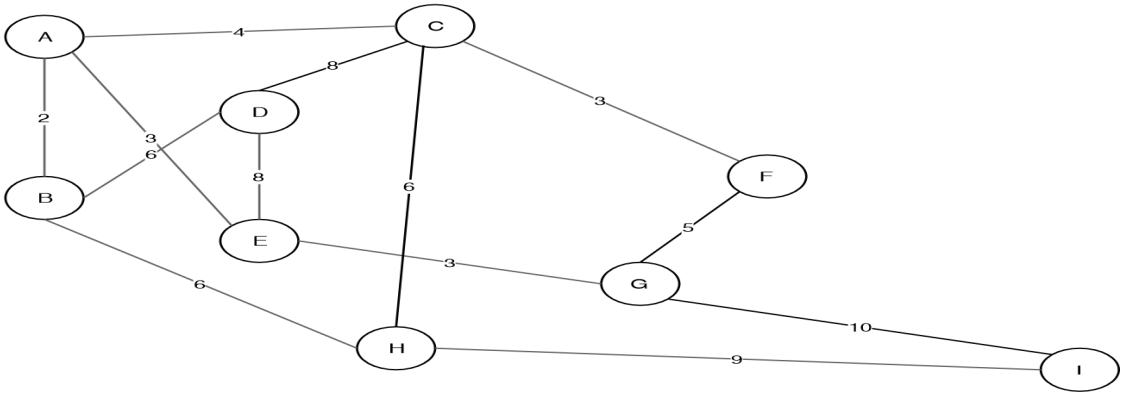
1- A-B edge is selected (2 weighted)
2- A-E edge is selected (3 weighted)
3- E-G edge is selected(3 weighted)
4- C-F edge is selected (3 weighted)
5- A-C edge is selected(4 weighted)
6- G-F edge is not selected because there is a cycle(5 weighted)
7- B-H edge is selected(6 weighted)
8- B-D edge is selected(6 weighted)

9- C-H edge is not selected because there is a cycle(6 weighted)
10- D-C edge is not selected because there is a cycle(8 weighted)
11- D-E edge is not selected because there is a cycle(8 weighted)
12- H-I edge is selected(9 weighted)
13- G-I edge is not selected there is a cycle (10 weighted)



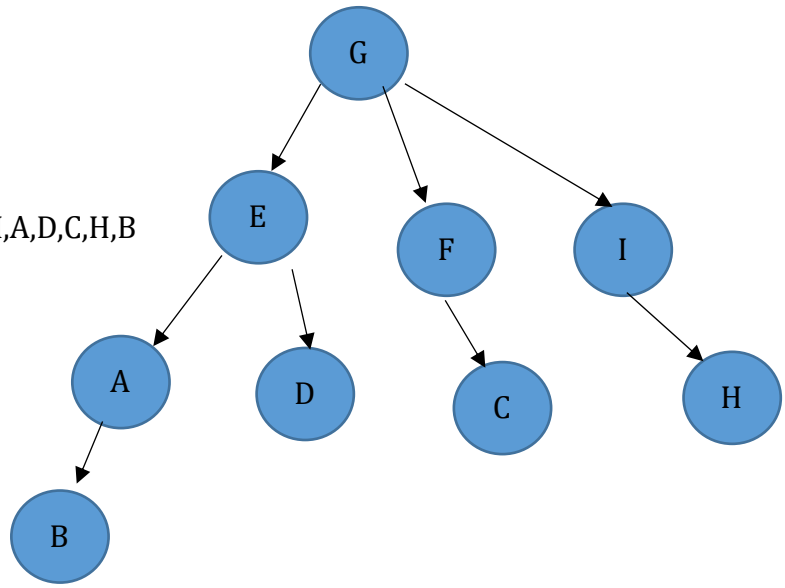
Question 5

Find shortest unweighted path from G to all other vertices for the graph in Figure 2. Use breadth-first search algorithm in your answer. Do NOT forget to show the trace.



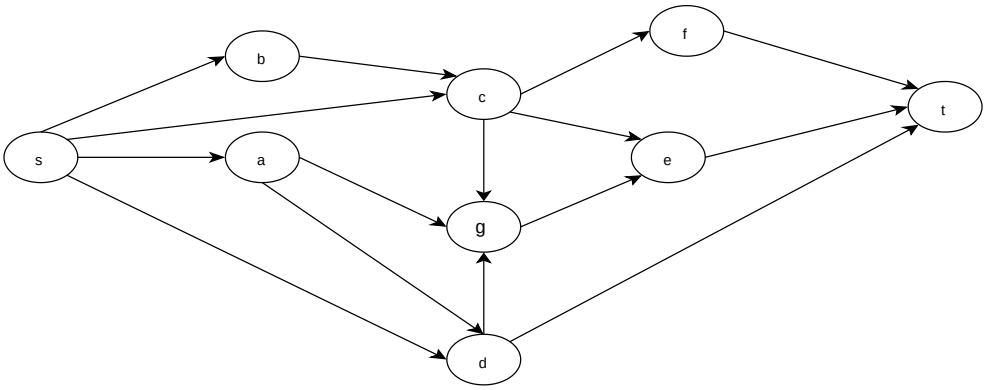
Weights are dismessed it is BFS.
We want to shortest path of G.Our root is G.First look the G's childs.E,F and I.Adding them by order picking.I look the childrens of E.A and D are childs of E adding them by order picking.Then look the F's child and C is child and add it.Look of the I's child and it is H and add it.We done with the G's childrens.We will now look the G's grandsons sons.Look the A 's childs.B is A's child add it the tree.Now we pointed all the nodes as visited.Completed.
(tree in the other page)

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



Quesiton 6

Find a topological ordering of the graph in Figure 3.



Select vertex with in degree 0.Print it out.Remove it.

1-Select S .Print S.Remove S.
2- Select B .Print B.Remove B.
3- Select A .Print A.Remove A.
4- Select C .Print C.Remove C.
5-Select F.Print F.Remove F.
6-Select D.Prınt D.Remove D.
7-Select G.Print G.Remove G.
8-Select E.Print E.Remove E.
9-Select T.Print T.Remove T.

Topological Order =S B A C F D G E T