Ahmad Abdullah i22-1609

CY-D

Design Analysis & Algorithms

Question #1

Iteration #1

Visited	a	b	c	d
cost	T	F	F	F
Parent	-1	a	a	a

Iteration #2

a	b	c	d	e	f
T	T	F	F	F	F
x	a	a	a	b	b

Iteration #3

a	b	c	d	e	f	i
T	T	F	F	T	F	F
0	3	5	1	3	2	4
	a	a	e	b	e	e

Iteration #4

a	b	c	d	e	f	h	i
T	T	F	F	T	F	F	F
0	3	2	1	3	2	5	4
	a	d	e	b	e	d	e

Iteration #5

a	b	c	d	e	f	g	h	i
T	T	T	T	F	F	F	F	F
0	3	2	1	3	2	4	5	4
	a	d	e	b	e	c	d	e

Iteration #6

a	b	c	d	e	f	g	h	i	j	k
T	T	T	T	T	T	T	F	F	F	F
0	3	2	1	3	2	4	3	4	5	6
	a	d	e	b	e	e	g	e	f	g

Iteration #9

i	j	k	L
T	F	F	F
4	3	6	5
e	i	g	i

Iteration# 10

i	j	k	L
T	T	F	F
4	3	6	5
e	i	g	i

Iteration #11

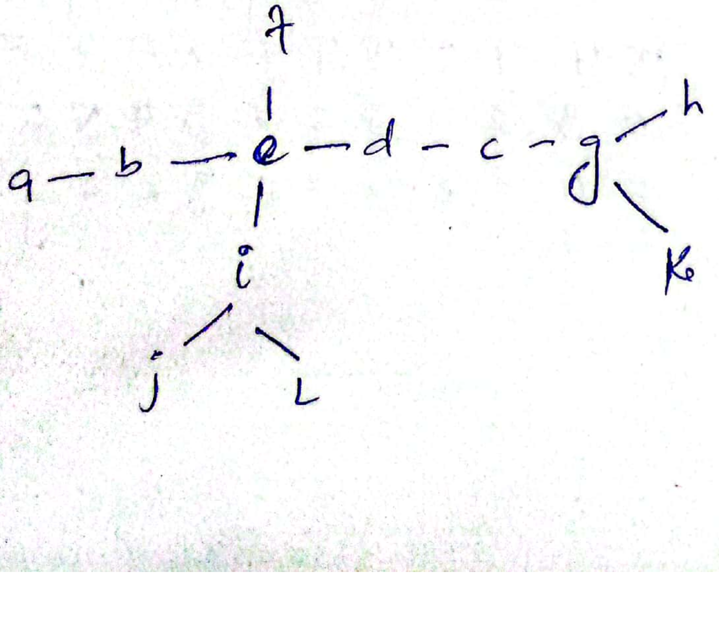
i	j	k	L
T	T	F	T
4	3	6	5
e	i	g	i

Iteration #12

i	j	K	L
T	T	T	T
4	3	6	5
e	i	g	i

Finally we have

	a	b	c	d	e	f	g	h	i	j	k	l
cost	0	3	2	1	3	2	4	3	4	3	6	5
Parent	x	a	d	e	b	e	c	g	e	i	g	i



Total cost 36

Scanned with CamScanner

Part 2: Prims algorithm assumes that graph is connected so it doesn't check connectivity. If the graph is not connected prims algorithm will only check and produce MST for the connected component containing the starting vertex

Part 3: No, not in general. Assume T has been generated by considering $V_1, V_2, \ldots$. A new vertex V_n be weighted edge (v_1 is root of T), if the weighted edge is smaller then weight of (v_1, v_2) in T, this would not be MST anymore,

Question #3 (B)

Dijikstra Algorithm assumes Path can only become heavier so that if you have paths $A \xrightarrow{3} B$ and $A \xrightarrow{5} C$ there's no way you can add an edge and get from A to B through C with weight less than 3. Thus it can not give correct result in negativity weight graph

Question # 2.

$a \xrightarrow{3} b$

$a \xrightarrow{5} c$

$a \xrightarrow{4} d$

$b \xrightarrow{6} f$

$b \xrightarrow{3} e$

$c \xrightarrow{4} g$

$c \xrightarrow{3} d$

$d \xrightarrow{1} h$

$e \xrightarrow{2} f$

$e \xrightarrow{5} i$

$f \xrightarrow{3} j$

$g \xrightarrow{6} h$

$g \xrightarrow{4} k$

$h \xrightarrow{3} k$

$h \xrightarrow{6} i$

$i \xrightarrow{7} l$

$i \xrightarrow{6} j$

$j \xrightarrow{5} l$

$k \xrightarrow{3} l$

$d \xrightarrow{1} e$

$c \xrightarrow{2} d$

$e \xrightarrow{3} f$

$a \xrightarrow{3} b$

$b \xrightarrow{3} e$

$g \xrightarrow{3} h$

$i \xrightarrow{3} j$

$e \xrightarrow{4} d$

$c \xrightarrow{4} g$

$e \xrightarrow{4} i$

$a \xrightarrow{4} c$

$d \xrightarrow{5} h$

$f \xrightarrow{5} j$

$i \xrightarrow{5} l$

$b \xrightarrow{5} f$

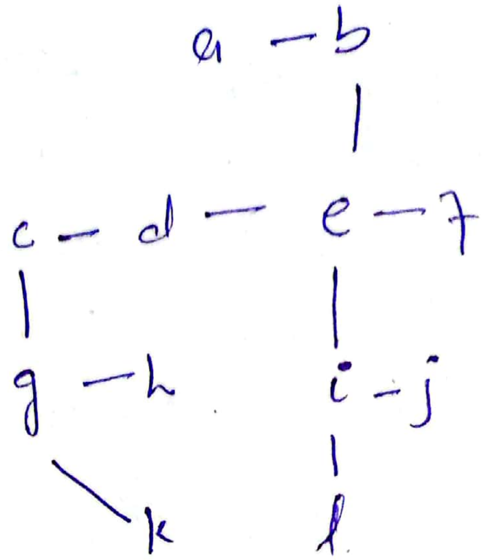
$g \xrightarrow{6} k$

$h \xrightarrow{6} i$

$h \xrightarrow{6} k$

$k \xrightarrow{7} l$

$i \xrightarrow{9} l$



MST cost: 36

Dijikstra:

Iteration #1

	A	D	H	j	i
	T	F	F	F	F
	0	1	5	-1	1
	-1	A	A	A	A

Iteration # 2

A	B	D	H	i	j
T	F	F	F	T	F
0	0	1	5	-1	1
-1	i	A	A	A	A

Iteration # 3

A	B	D	G	H	i	J
T	T	F	F	F	T	F
0	0	1	5	5	-1	1
-1	i	A	B	A	A	A

Iteration #4

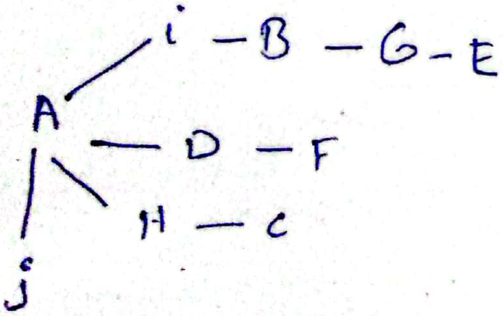
A	B	D	F	G	H	i	J
T	T	T				T	
0	0	1	4	5	5	-1	1

⋮

Iteration # 8

A	B	C	D	E	F	G	H	i	j
T	T	F	T	F	T	T	T	T	T
0	0	6	1	8	4	5	5	-1	1
	i	H	A	G	D	B	A	A	A

No change in iteration 9 & 10

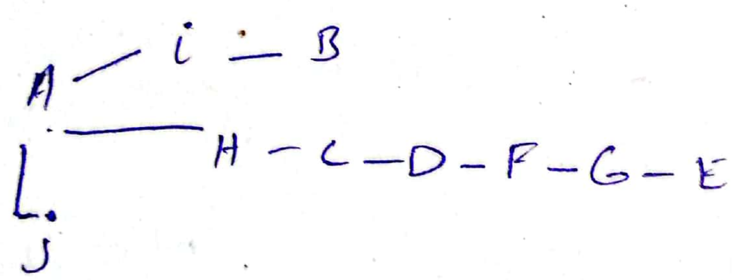


(ii) Belman Ford

	A	B	C	D	E	F	G	H	I	J
1	0	0	6	-4	8	-1	5	5	-1	-1
2	0	0	6	-4	8	-1	5	5	-1	-1
3	0	0	6	-4	4	-1	5	5	-1	-1
4										
5										
6										
7										
8										
9										

No change.

with Belman Ford we get



Question #4

LCS (s1 , s2)

m = s1.length()

n = s2.length.

{

return res

s1 = "abcdefefgh"

s2 = "bcdeggfefgg"

	x	b	c	d	e	g	g	f	e	f	g	g
x	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0	0	0	0	0	0
b	0	1	0	0	0	0	0	0	0	0	0	0
c	0	0	2	0	0	0	0	0	0	0	0	0
d	0	0	0	3	0	0	0	0	0	0	0	0
e	0	0	0	0	4	0	0	0	1	0	0	0
f	0	0	0	0	0	0	0	1	0	2	0	0
e	0	0	0	0	1	0	0	0	2	0	0	0
f	0	0	0	0	0	0	0	1	0	3	0	0
g	0	0	0	0	0	1	1	0	0	0	4	1
h	0	0	0	0	0	0	0	0	0	0	0	0

Largest comment string length is : 4

Question #5

Function countingSums(int n)
{ int dp[n] = {0}
dp{0} = 1
{
|
|
return dp[n]-1;
}

For n = 50
returns 204225

for n = 10

1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	3	3	4	5	5	6
3	1	1	2	3	4	5	7	10	12	14
4	1	1	2	3	5	6	9	15	18	23
5	1	1	2	3	5	7	10	18	23	30
6	1	1	2	3	5	7	11	20	26	35
7	1	1	2	3	5	7	11	21	28	38
8	1	1	2	3	5	7	11	22	29	40
9	1	1	2	3	5	7	11	22	30	41
10	1	1	2	3	5	7	11	22	30	42

result = 41