Home Assignment 2

Advanced Data Structures

① **Algorithm for Binary Search**

```
Algorithm BinarySearch(a,s,e,k)
{
 if(s==e)
 {
  if(key==a[s])
    return s;
  else
    return -1;
 }
else //problem is not small problem
{
mid=(s+e)/2
if(key==a[mid])
    return mid;
else if(key<mid)
{
BinarySearch(a,s,mid-1,k)
}
else
{
BinarySearch(a,mid+1,e,k)
}
}
```

Time Complexity Analysis

i, **Best Case**: If the search element is in the middle position or If the array contains single element

• $O(1)$ is the best case time complexity

ii, **Worst Case**

$$T(n) = T(n/2) + c$$

T(n/2) is the time taken by recursive call of left of mid or right of the mid

$T(n) = T(n/2) + c$ $\boxed{T(n/2) = T(n/4) + c}$

$= T(n/4) + 2c$

$= T(n/8) + 3c$

Similarly

$= T(n/16) + 4c$

$T(n) = T(\frac{n}{2^i}) + ic$

We know that, $n = 2^i$

$\boxed{i = \log_2 n}$

$T(n) = T(1) + c \cdot \log_2 n$

$= O(1) + c \log_2 n$

$= O(\log_2 n)$

=) Time complexity is O(nlogn)

Ex:- Search 23

0	1	2	3	4	5	6	7	8	9
2	5	8	12	16	23	38	56	72	91

$\frac{0+9}{2} = 4.5$ mid = 4

0	1	2	4	5	6	7	8	9
2	5	8	12	23	38	56	72	91

key = 23

key > mid.

$\frac{5+9}{2} = \frac{14}{2} = 7$

5	6	7	8	9
23	38	56	72	91

mid = 7 ⇒ 56

key < 56 (mid)

$\frac{5+7}{2} = \frac{12}{2} = 6.$

5	6	|	7
23	38		

mid = 38 (6)

key < mid

$\frac{5+6}{2} = \frac{11}{2} = 5..$

5	|	6
23		38

mid = 23

key = mid = 23 ✓

key is found in 5th location

(10)

Item	A	B	C	D
Profit	280	100	120	120
Weight	40	10	20	24

Given $W=60$

$U=60$

A:- Decreasing order of P_i/W_i Ratio

$$\frac{P_i}{W_i} \Rightarrow \quad \overset{A}{\frac{280}{40}} \quad \overset{B}{\frac{100}{10}} \quad \overset{C}{\frac{120}{20}} \quad \overset{D}{\frac{120}{24}}$$

$$\Rightarrow \quad 7 \quad 10 \quad 6 \quad 5$$

So the order is B A C D

B A C D

$x_i = [0 \quad 0 \quad 0 \quad 0]$

Now for $i=1$ (B)

$x_1 \rightarrow w[i] \geq 0$

$w[i] \geq 60$ $\qquad w[i] = B \rightarrow 10.$

$10 \geq 60$ ×.

$x_1 = 1.0 \quad U = U - w[i]$

$= 60 - 10$

$U = 50$

For $i=2$ (A)

$x_2 \rightarrow w[2] \geq 0$

$40 \geq 50$ ×

$x_2 = 1.0 \quad U = U - w[2]$

$= 50 - 40$

$= 10.$

For $i=3$ (C)

$\Rightarrow X_3 \Rightarrow W[3] \geq U$

$20 \geq 10$ ✓

then

$X[i] = U/W[i]$

$X[3] = \frac{10}{20}$

$X_3 = \frac{1}{2}$

For $i=4$ (D)

$X_4 \rightarrow W[4] \geq U$

$24 \geq 10$ ✓

$X[4] = \frac{0}{24}$

$X_4 = 0$

$X_i = [1, 1, 1/2, 0]$

$\sum W_i X_i = (1*40 + 1*10 + \frac{1}{2}*20 + 0*24)$

$= 40 + 10 + 10 + 0$

$= 60$

Profit

$\sum P_i X_i = (1*280 + 1*100 + \frac{1}{2}*120 + 0*120)$

$= 280 + 100 + 60 + 0$

$= 440$

(15) Find the minimum cost spanning tree using prims and kruskals algorithms.

A:- **Kruskals algorithm**

Edgencency Matrix

	a	b	c	d	e	f	g	h
a	0	2	0	0	10	8	7	0
b	2	0	5	7	5	0	0	4
c	0	5	0	9	0	1	0	0
d	0	7	9	0	2	3	0	0
e	10	5	0	2	0	6	0	0
f	8	0	1	3	6	0	3	0
g	7	0	0	0	0	3	0	11
h	0	4	0	0	0	0	11	0

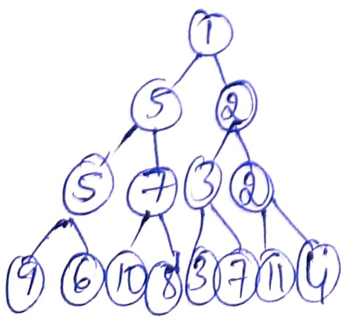
h a b

g e c

f d

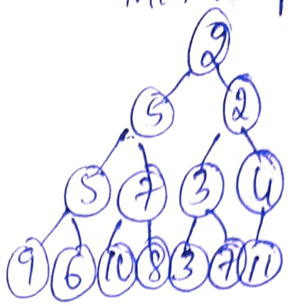
Min heap

2, 5, 1, 9, 7, 3, 2, 6, 5, 10, 8, 3, 7, 11, 4



V	a	b	c	d	e	f	g	h
P	-1	-1	-1	-1	-1	-1	-1	-1

Min heap



Iteration-1

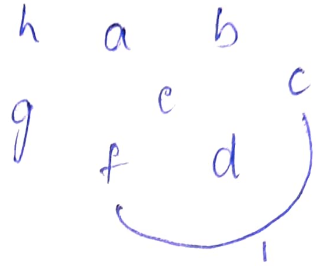
delete 1 → c-f(edge)

j=find(c) k=find(f)

=c =f

P[c]=f

v	a	b	c	d	e	f	g	h
p	-1	-1	f	-1	-1	-1	-1	-1



Iteration-2

delete 2

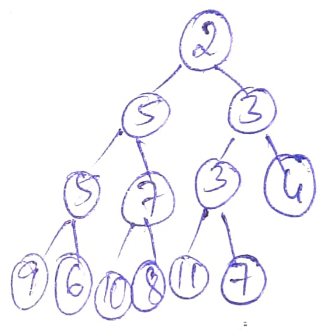
a-b

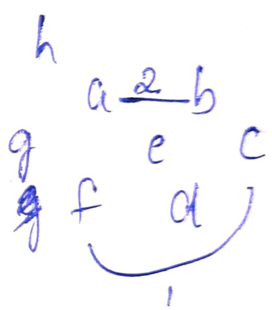
j=find(a) k=find(b)

=a =b

P[a]=b

Min heap





v	a	b	c	d	e	f	g	h
p	b	-1	f	-1	-1	-1	-1	-1

Iteration-3

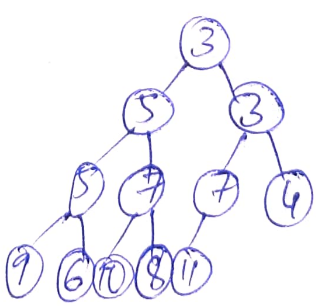
delete 2.

Edge d-e

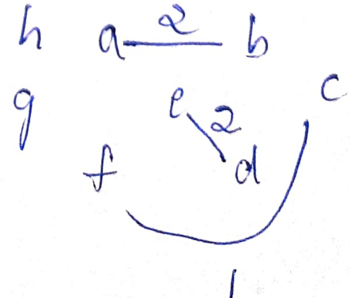
j=find(d) k=find(e)

=d =e

P[d]=e



v	a	b	c	d	e	f	g	h
p	b	-1	f	e	-1	-1	-1	-1



Iteration-4

delete 3

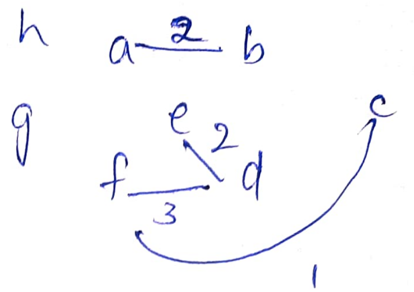
Edge d-f

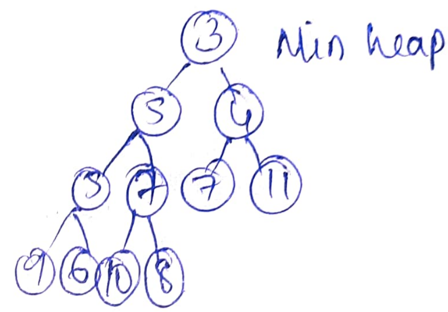
j = find(d) k = find(f)

= e   = f

P[e] = f

v	a	b	c	d	e	f	g	h
p	b	-1	f	e	f	-1	-1	-1





Iteration-5

delete 3

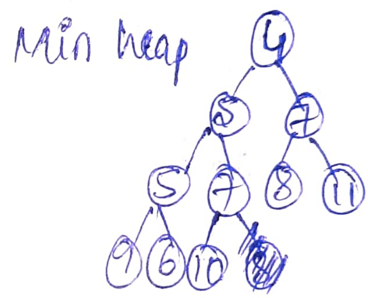
Edge f-g

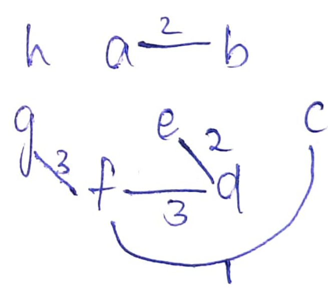
j = find(f) k = find(g)

= f   = g

P[f] = g

v	a	b	c	d	e	f	g	h
p	b	-1	f	e	f	g	-1	-1





Iteration-6

delete 4

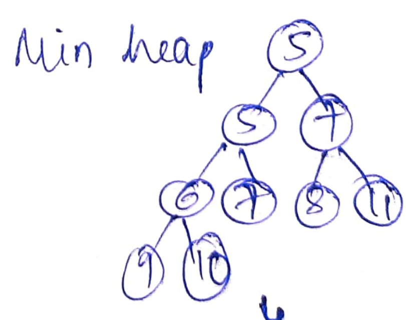
Edge b-h

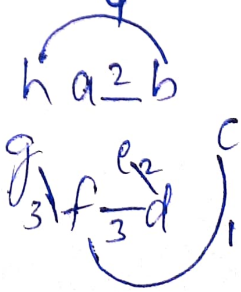
j = find(b) k = find(h)

= b   g = h

P[b] = ~~g~~, P[b] = h

v	a	b	c	d	e	f	g	h
p	b	h	f	e	f	g	-1	~~-1~~ g





Iteration-7

delete 5

Edge b-c

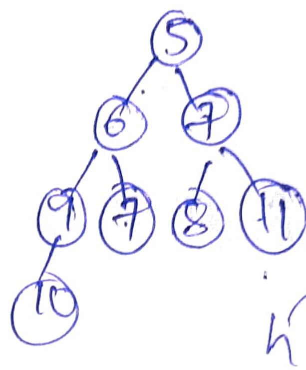
j= find(b)   k=find(c)

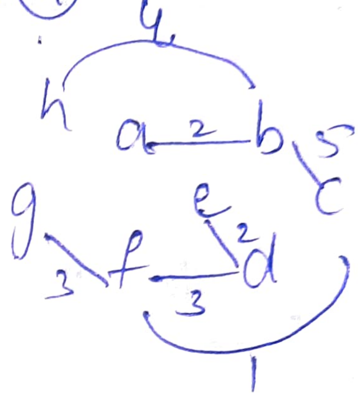
= h   = g

p[h]=g.

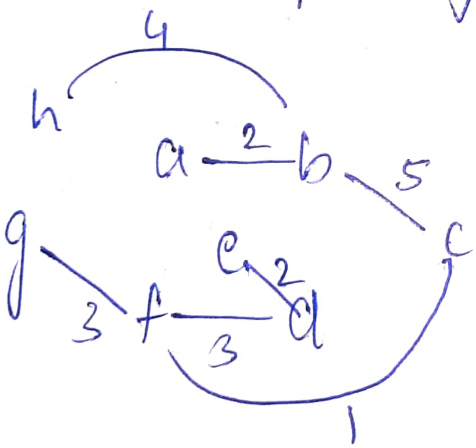
v	a	b	c	d	e	f	g	h
p	b	h	f	e	f	g	-1	g

Min heap





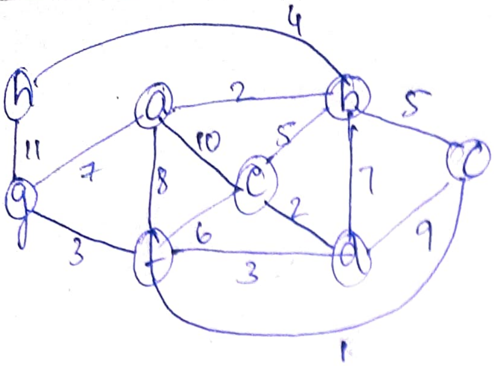
Minimum Cost Spanning tree is (20)



∴ The minimum cost for the given spanning tree is

$4+5+3+3+2+2+1=20$.

(15)



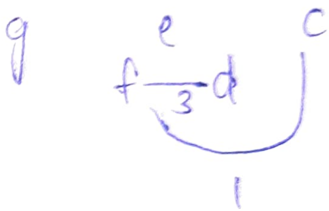
Prims algorithm

The min value edge is f-c

c-f, Iteration-1

v	d[c]	d[f]	near[v]	cost(v,near(v))
a	∞	8	f	8
b	5	∞	c	5
c	∞	∅	~~f~~	
d	9	3	f	3
e	∞	6	f	6
f	1	∞	~~c~~	
g	∞	3	f	3
h	∞	∞	—	∞

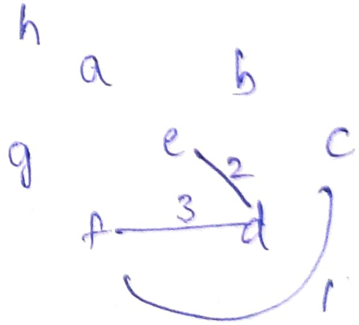
d,near(d)
d,f ⇒ 3.



Iteration-2

v	cost	dist[d]	near[v]	cost
a	8	∞	f	8
b	5	7	c	5
c	—	—	—	—
d	—	—	—	—
e	6	2	d	2
f	—	—	—	—
g	3	∞	f	3
h	∞	∞	—	∞

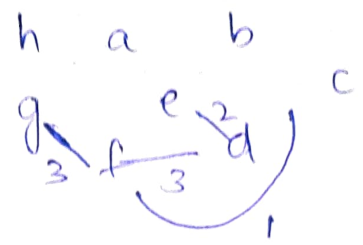
e,near[e]
e,d ⇒ 2



Iteration 3

v	cost	dist[e]	near[v]	cost
a	8	10	f	8
b	5	5	c	5
c	—	—	—	—
d	—	—	—	—
e	—	—	—	—
f	—	—	—	—
g	3	∞	f	3
h	∞	∞	∞	∞

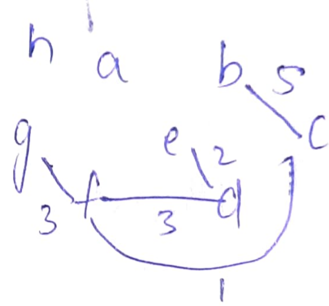
g, near(g)
g, f ⇒ 3



Iteration 4

v	cost	dist[g]	near[v]	cost
a	8	7	g	7
b	5	∞	c	5
c	—	—	—	—
d	—	—	—	—
e	—	—	—	—
f	—	—	—	—
g	—	—	—	—
h	∞	11	g	11

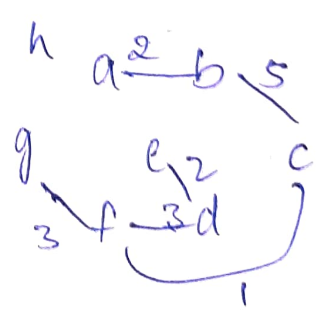
b, near(b)
b, c ⇒ 5



Iteration 5

v	cost	dist(b)	near[v]	cost
a	7	2	b	2
b	—	—	—	—
c	—	—	—	—
d	—	—	—	—
e	—	—	—	—
f	—	—	—	—
g	—	—	—	—
h	11	4	b	4

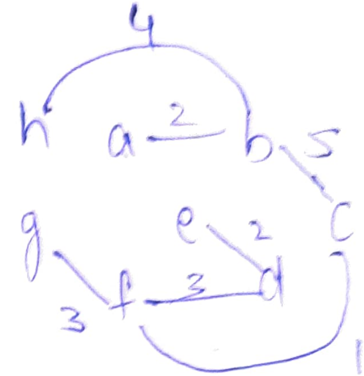
a, near(a)
a, b ⇒ 2



Iteration-6

v	cost	dist[a]	nar[v]	cost
a	—	—	—	—
b	—	—	—	—
c	—	—	—	—
d	—	—	—	—
e	—	—	—	—
f	—	—	—	—
g	—	—	—	—
h	4	∞	b	4

h, near[h]
h, b => 4



∴ The minimum cost is 20 using prims algorithm

(20)

Job	J_1	J_2	J_3	J_4	J_5	J_6
Deadline	5	2	4	3	3	1
Profit	20	40	5	15	10	8

Job with height profit is J_2

40
J_2

0 1

$d[J_2] \geq 1$
$2 \geq 1$ ✓

Job with next highest profit is J_1

40	20
J_2	J_1

0 1 2

$P = 40 + 20$
$= 60$

$d[J_1] \geq 2$
$5 \geq 2$ ✓

Job with next highest profit is J_4

40	20	15
J_2	J_1	J_4

0 1 2 3

$d[J_4] = 3$
$3 \geq 3$ ✓

Profit $= 40 + 20 + 15$
$= 75$

Job with the next highest profit is J_5

$d[J_5] \geq 4$
$3 \geq 4$ X

10	40	15	20
J_5	J_2	J_4	J_1

0 1 2 3 4

$\therefore d[J_4] = 3$ can't be moved
$d[J_1] = 5$ can be moved.
$d[J_2] = 2$ can be moved

$40 + 20 + 10 + 15$
$= 85$

Job with next highest profit is J_6

$d[J_6] = 1$

J_2 and J_5 cant be moved $P[J_6] = 8$

J_5 can't be placed anywhere

J_6 can't be considered

as $P[J_5] > P[J_6]$

$10 > 8$

Next Job is J_3

$d[J_3] \geq 5$

$4 \geq 5$ ×

J_1 can be moved to 4, 5

$d[J_1] = 5$

J_5	J_2	J_4	J_3	J_1

0 1 2 3 4 5

$P = 85 + 5$

$= 90$

∴ Job sequence with max profit is

$J_5 - J_2 - J_4 - J_3 - J_1$

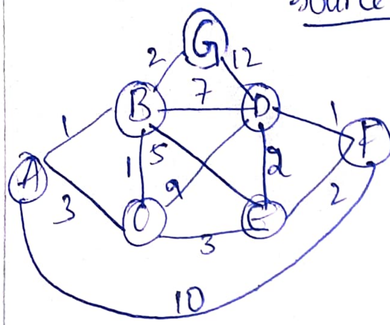
(or)

$J_2 - J_5 - J_4 - J_3 - J_1$

with a profit of 80.

25

Source A



City	Dist
A	0
B	
C	
D	
E	
F	
G	

update the distance →

City	Dist
A	0 ✓
$\boxed{B}$	$\boxed{1}$ min
C	3
D	∞
E	∞
F	10
G	∞

distance from A considering B as intermediate

City	Dist
A	0 ✓
B	1 ✓
$\boxed{C}$	$\boxed{2}$ ✓
D	8
E	6
F	10
G	3

min[dist(c), dist(b)+(c,b)] = [3, 1+1] = 2
min[dist(d), dist(b)+(d,b)] = [∞, 1+7] = 8
min(dist(e), dist(b)+(e,b)] = [∞, 1+5] = 6
min[d(f), dist(b)+(f,b)] = [10, 1+∞] = 10
min[d(g), d(b)+(g,b)] = [∞, 1+2] = 3

distance from A considering B-C as intermediate

City	Dist
A	0 ✓
B	1 ✓
C	2 ✓
D	8
E	5
F	10
$\boxed{G}$	$\boxed{3}$ ✓

min[dist(d), dist(c)+(d,c)] = [8, 3+9] = 8
min[d(e), d(c)+(e,c)] = [6, 2+3] = 5
min[d(f), d(c)+(f,c)] = [10, 3+∞] = 10
min[d(g), d(c)+(g,c)] = [3, 3+∞] = 3

B-C-G intermediates

City	Dist
A	0 ✓
B	1 ✓
C	2 ✓
D	8
$\boxed{E}$	$\boxed{5}$ ✓
F	10
G	3 ✓

min[d(d), d(g)+(g,d)] = [8, 3+12] = 8
min[d(e), d(g)+(g,e)] = [5, 3+∞] = 5
min[d(f), d(g)+(g,f)] = [10, 3+∞] = 10

B-C-G-E as intermediates

City	dist
A	0 ✓
B	1 ✓
C	2 ✓
$\boxed{D}$	$\boxed{7}$ ✓
E	5 ✓
F	7
G	3 ✓

$\min[d(d), d(e)+l(d,e)] = [8, 5+2] = 7$

$\min[d(f), d(e)+l(f,e)] = [10, 5+2] = 7$

D as intermediate

City	dist
A	0
B	1 ✓
C	2 ✓
D	7 ✓
E	5 ✓
$\boxed{F}$	$\boxed{7}$ ✓
G	3 ✓

$\min[d(f), d(d)+l(f,d)] = [7, 7+1] = 7$

Final:

City	dist from A	Path
A	0	0
B	1	A-B = 1
C	2	A-B-C = 1+2 = 3
D	7	A-B-C-~~A~~-E-D = 1+1+3+2 = 7
E	5	A-B-GE = 5
F	7	A-B-C-E-F = 7
G	3	A-B-G = 3