208W1A1263

Home Assignment-2

1) Design an algorithm for quick sort using divide and conquer strategy and estimate the time complexity of the designed algorithm in best, average and worst cases. Trace algorithm with example.

A)
```
Algorithm quicksort (a,s,e,n){
    if(s<e){
    j=partition(s,e);
    quick sort(a,s,j-1,n/2);
    quick sort (a,j+1,e,n/2);
    }
}
Algorithm Partition (s,e){
    pivot=s, i=s+1, j=e;
    while (i<j) {
    while (a[i] < pivot) {
        i++;}
    while(a[j] > pivot){
        j--;}
    if (i<j){
        swap(a[i],a[j])
    }
    }
    swap(a[j],pivot)
    return j;
}
```

Time complexities

1) Best case:-

$T(n) = 2T(n/2)+cn = 2[2T(n/4)+\frac{cn}{2}]+cn$

$= 2^3+(n/2^3)+3cn$

$= 2^i+(n/2^i)+icn$

$\frac{n}{2^i}=1 \Rightarrow n=2^i \Rightarrow i=\log_2 n$

$= 2^{\log_2 n^2}+\left(\frac{n}{2^{\log_2 n}}\right)+\log_2 n\, c_1$

$= nT(1) + \log_2 n * c * n$

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

ii) **worst case**

$T(n) = T(n-1) + c_n = T(n-2) + c(n-1) + c_n$

$= T(n-3) + c(n-2) + c(n-1) + c_n$

$= T(n-4) + c(n-3) + c(n-2) + c(n-1) + c_n$

To get $T(1)$ put $T(n) = T(n-(n-1))$

$= T(n-(n-1)) + c.2 + c.3 + c.4 + \dots + c.n$

$= T(1) + c(2+3+\dots+n)$

$= 1 + c\left(\frac{n(n+1)}{2} - 1\right) = 1 + \frac{cn^2}{2} + \frac{cn}{2} - 1$

$= O(n^2)$

(iii) **Average case**

$T(n) = \left[\sum_{i=1}^{n} T(i-1) + T(n-i)\right] * \frac{1}{n} + n + 1 \rightarrow ①$

$n(T(n)) = \sum_{i=1}^{n} [T(i-1) + T(n-i)] + n(n-1)$

$n * T(n) = T(0) + T(n-1) + T(1).(n-2) + T(2)(n-3) + \dots + (n-2) + T(1) + T(0) + n(n+1)$

$n * T(r) = 2[T(0) + \dots + T(n-1)] + n(n+1) \rightarrow ②$

explore 'n' with $(n-1)$ in the above eqn ②

$(n-1) * T(n-1) = 2[T(0) + \dots + T(n-2)] + (n-1)n \rightarrow ③$

Subtract ② with 3 => 2-3

$n * T(n) - (n-1) * T(n-1) = 2T(n-1) + n(n+1) - (n-1)n$

$n(T(n)) - (n-1)T(n-1) = 2T(n-1) + 2n$

$nT(n) = 2T(n-1) + (n-1) + T(n-1)) + 2n$

divide the above equation $n(n+1)$

$\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2}{n+1}$

now apply substitution method

$\frac{T(n)}{n+1} = \frac{T(n-2)}{n-1} + \frac{2}{n} + \frac{2}{n+1}$

$$\frac{T(n)}{n+1} = \frac{T(n-3)}{n-2} + \frac{2}{n-1} + \frac{2}{n} + \frac{2}{n+1},$$

$$\frac{T(n)}{n+1} = \frac{T(n-4)}{n-3} + \frac{2}{n-2} + \frac{2}{n} + \frac{2}{n+1},$$

generalisation

$$\frac{T(n)}{n+1} = \frac{T(n-(n-1))}{2} + \frac{2}{3} + \frac{2}{4} + \frac{2}{5} + \cdots + \frac{2}{n+1},$$

$$\frac{T(n)}{n+1} = \frac{1}{2} + 2\left[\frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n+1}\right]$$

$$\frac{T(n)}{n+1} = \frac{1}{2} + 2\left[\sum_{k=3}^{n+1} \frac{1}{k}\right]$$

The entire sigma notation can be written as $\int_{k=2}^{n+1} 1/k$

$$\frac{T(n)}{n+1} = 1/2 + 2\left[\log k\right]_2^{n+1}$$

$$\frac{T(n)}{n+1} = 1/2 + 2(\log(n+1) - \log 2)$$

$$\frac{T(n)}{\cancel{n+1}} = \frac{n+1}{2} = 2(n+1)(\log(n+1) - 2(n+1)\log 2)$$

$$T(n) = O(n\log n)$$

Example

54, 26, 93, 17, 77, 31, 44, 55, 20
↗ ↑ ↑
pivot i j

54, 26, 93 17 77 31 44 55 20
↑ ↑ ↑
pivot i j

54 26 20 17 44 31 77 55 93
↑ ↑
pivot i j

31 26 20 17 14 $\textcircled{54}$ 77 55 93
↑ j ↑ ↑
pivot i j

31 26 20 17 44 $\boxed{54}$ 77 55 93
↑ j i pivot j i
pivot

17   26   20 $\boxed{31}$ $\boxed{44}$ $\boxed{54}$ $\boxed{55}$ $\boxed{77}$ $\boxed{93}$

pivot   i   j

$\boxed{17}$ 26   20 $\boxed{31}$ $\boxed{44}$ $\boxed{54}$ $\boxed{55}$ $\boxed{77}$ $\boxed{93}$

pivot   ij

$\boxed{17 | 20 | 26 | 31 | 44 | 54 | 55 | 77 | 93}$

2)

objects	1	2	3	4	5	6	7
profits	10	5	15	7	6	18	5
weights	2	3	5	7	1	4	3

solve the problem as knapsack problem

a) P/w   5   1.66   3   1   6   4.5   3

x_i values $[1, 2/3, 1, 0, 1, 1, 1]$, Given, $r=15$

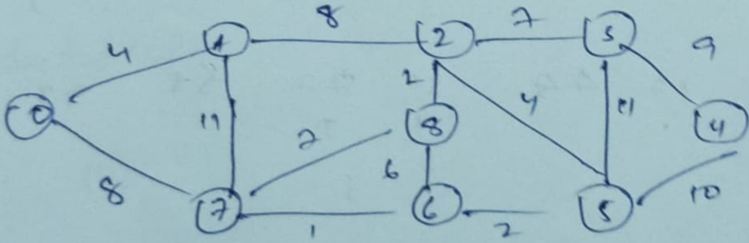
$\Sigma w_i x_i = 2+2+5+1+4+3 = 15 \le r$

$\Sigma p_i x_i = 10+10/3+15+6+8+3 = \frac{10}{3}+52 = 166/3$

$\Sigma p_i x_i = 55.33$

∴ optimal solution is $(1, 2/3, 1, 0, 1, 1, 1)$ with profit 55.33

3) Find the minimum cost spanning tree using prisms and kruskal Algorithm

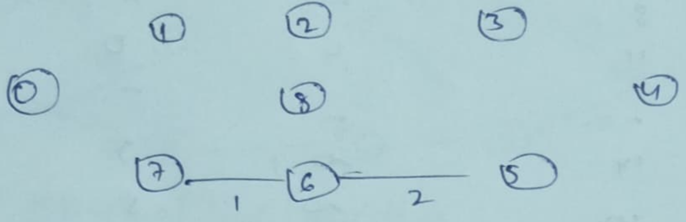


using prisms Algorithm

Iteration 1 :- minimum cost is 1, 80, 6 and 7 selected.

vertex	dist[6]	dist[7]	near	mindist
0	∞	8	7	8
1	∞	11	7	11
2	∞	∞	–	∞
3	∞	∞	–	∞
4	∞	∞	–	∞
5	2	∞	–	2
6	∞	1	6	–
7	1	80	6	–
8	6	7	6	6

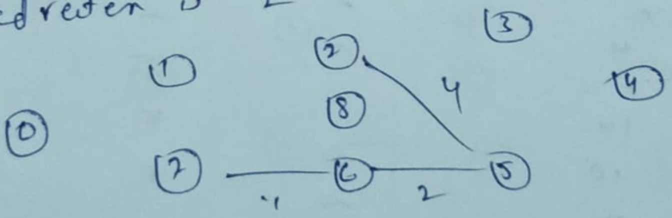
selected vertex is 5



Iteration 2 :

vertex	cost [v, near(v)]	dist [5]	near	mindist
0	8	∞	7	8
1	11	∞	7	11
2	∞	4	5	4
3	∞	14	5	14
4	∞	10	5	10
5	2	–	–	–
6	–	–	–	–
7	–	–	–	–
8	6	∞	6	6

selected vertex is 2

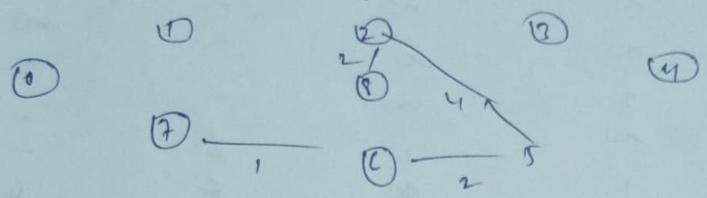


Iteration 3 :

vertex	cost [v, near[v]]	dist 2	near	mindist
0	8	∞	2	8
1	11	8	2	8
2	4	7	–	–
3	14	2	2	7
4	10	∞	5	10

5	-	-	-	-
6	-	-	-	-
7	-	-	-	-
8	6	2	2	2

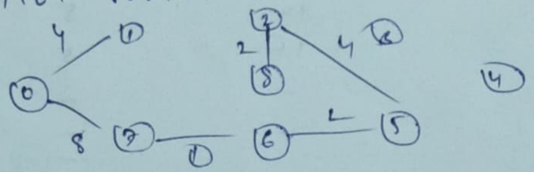
selected vertex is 8



Iteration 4:

vertex	cost [v, neal [v]]	dist (0)	neal	mind!, A
0	8	-	-	-
1	11	4	0	4
2	-	-	-	-
3	14	∞	5	14
4	10	∞	5	10
5	-	-	-	-
6	-	-	-	-
7	-	-	-	-
8	-	-	-	-

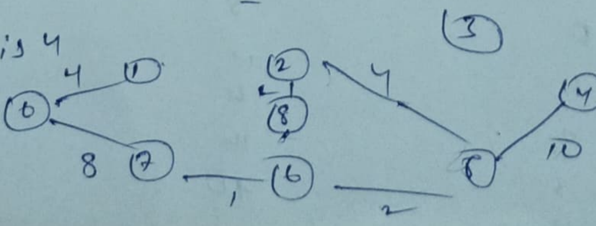
selection vertex



Iteration 6:

vertex	cost [v, neal[v]]	dist[1]	neal	mindist
0	-	-	-	-
1	4	-	-	-
2	-	-	-	-
3	14	∞	5	14
4	10	∞	5	10
5	-	-	-	-
6	-	-	-	-
7	-	-	-	-
8		-	-	-

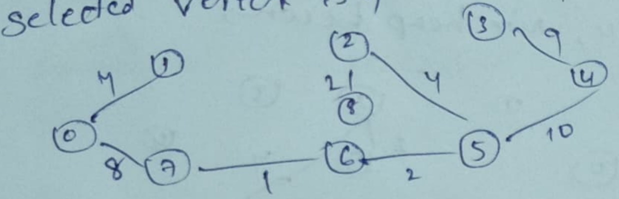
selected vertex is 4



Iteration 7

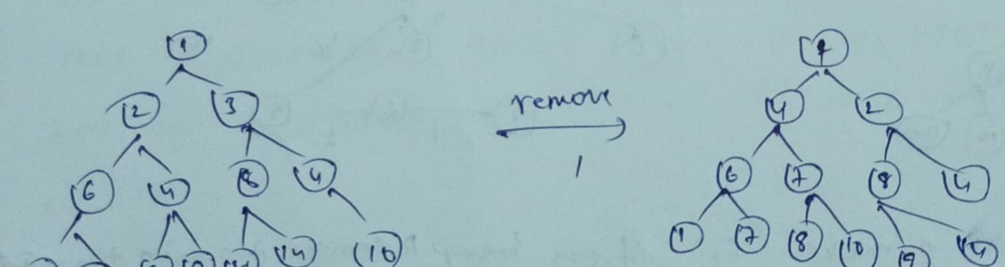
vertex	cost [v,near[v]]	dist[u]	near	mindist
0	–	–	–	–
1	–	–	–	–
2	14	–	–	–
3	10	9	4	9
4	–	–	–	–
5	–	–	–	–
6	–	–	–	–
7	–	–	–	–
8	–	–	–	–

selected vertex is 4

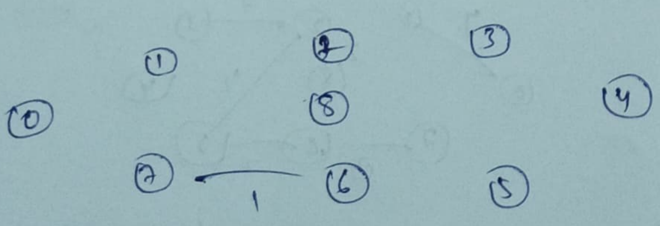


Using Krushal Algorithm

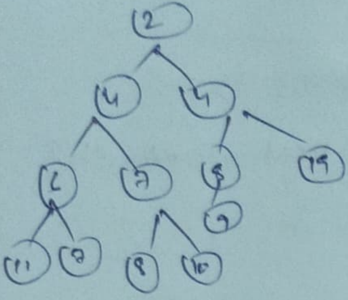
4, 11, 8, 8, 7, 1, 3, 6, 2, 7, 4, 9, 14, 10 on the costs given.
Construct a min heap costs we get heap as follows

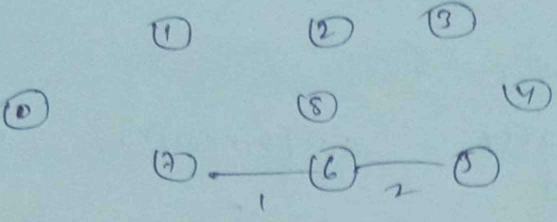


Iteration 1:

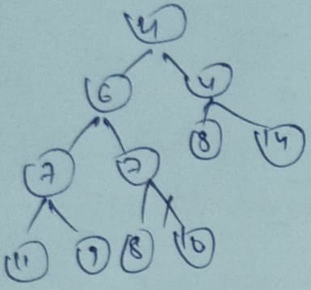


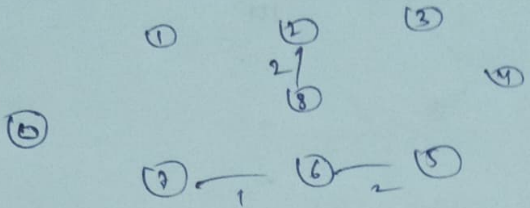
Iteration 2
Remove 2 from min heap it becoms.



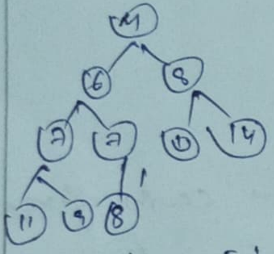


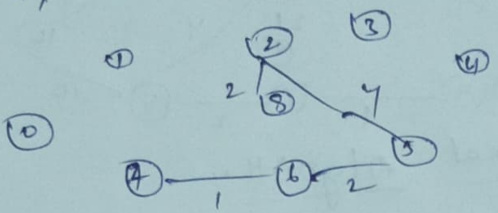
Iteration 3: Remove 2, Min heap becoms



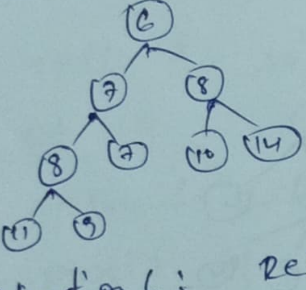


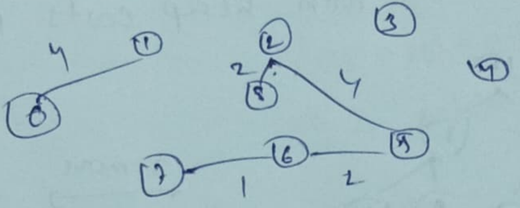
Iteration 4: Remove 4, Min heap becoms



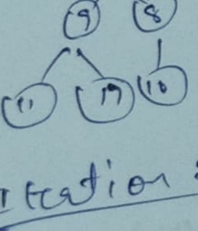


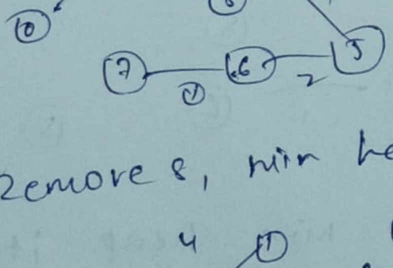
Iteration 5:



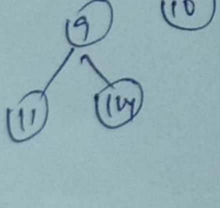


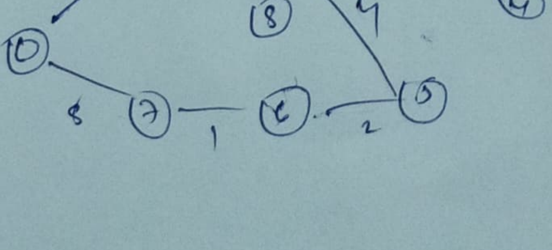
Iteration 6: Remove 6,7 from heap Remove 7 again to avoid loop



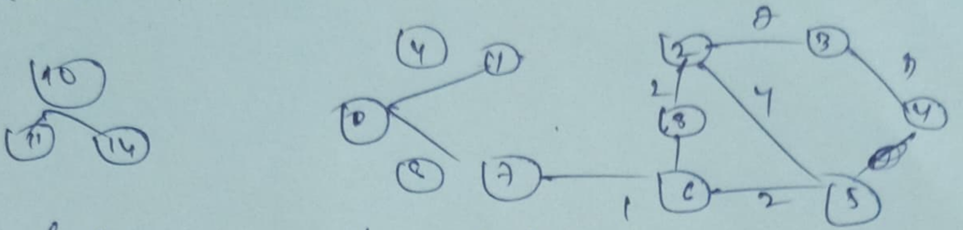


Iteration 7: Remove 8, min heap becoms





Iteration 8: Remaining 8 forms a loop in spanning tree. So remove 9. By remaining adding 10, 11, 14 forms loops. minimum cost spanning tree is



5) Find the optimal solution for the above job sequence units deadline problem.

A)

Job considered	sustarsign	solution	profit
J1	(2,3)	J1	35
J2	(2,3) (3,4)	J1, J2	35+30
J3	(1,2) (2,3) (3,4)	J1, J4, J3	35+30+25
J4	(5,1) (1,2) (2,3) (3,4)	J1, J2, J3, J4	35+30+25+20

As the deadlines for J5, J6, J7 are completed they are discarded total profit is 35+30+25+20=110 optimal solution is J1 → J2 → J3 → J4 with a profit of 110.

optimal solution is J1 → J2 → J3 → J4 with a profit of 110.