

UNIT-II

Greedy method:

Job Sequencing with Deadlines:

Consider a sequencing problem, where there are 4 jobs, those are J_1, J_2, J_3 , and J_4 .

	J_1	J_2	J_3	J_4
Profit	30	15	10	25
Deadline	2	1	2	1

Step 1: Ascending order of jobs based on profit.

(J_1, J_4, J_2, J_3)

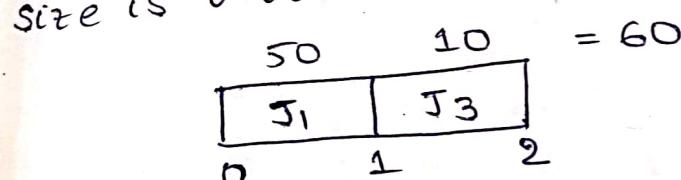
and their corresponding deadlines.

(2, 1, 1, 2)

Step 2: Consider highest deadline and find \max profit we get in that deadline.

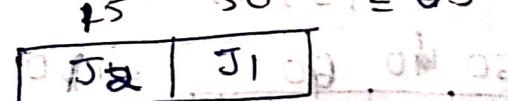
Step 2: Draw the gantt chart based on deadlines of the jobs.

Here, the deadline is 2. Then, the gantt chart size is 0 to 2.

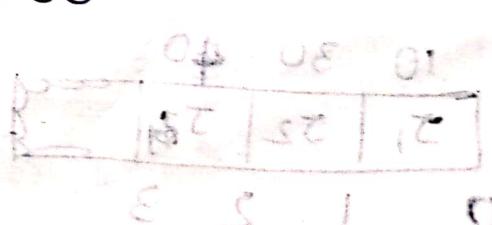
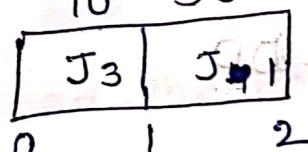


The combination of jobs J_1 & J_3 we can get

the profit 60.



$$10 + 50 = 60$$



$$\begin{array}{|c|c|} \hline 25 & 50 \\ \hline J_4 & J_1 \\ \hline \end{array} = 75$$

$$\begin{array}{|c|c|} \hline 25 & 10 \\ \hline J_4 & J_3 \\ \hline \end{array} = 35$$

$$\begin{array}{|c|c|} \hline 25 & 10 \\ \hline J_2 & J_3 \\ \hline \end{array} = 35$$

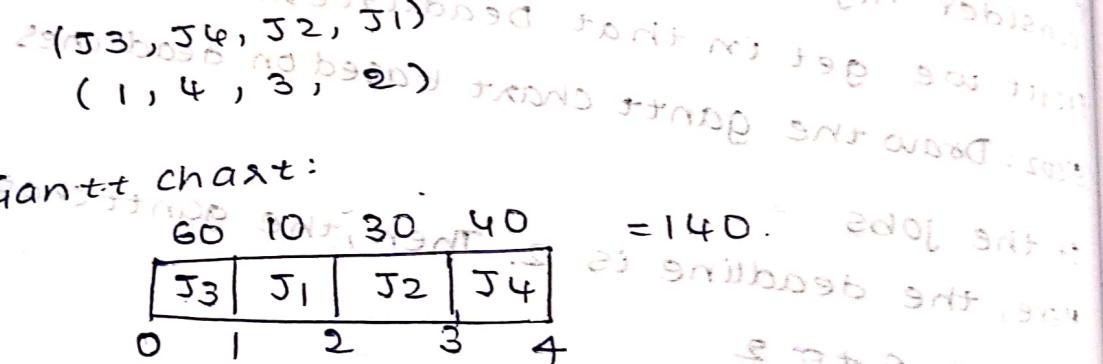
of both is 2

\therefore we can get the maximum profit by selecting the sequence of jobs by considering deadlines.

max profit we can get = 75

Q. consider a sequencing problem where there are 4 jobs having the profit of P(10, 30, 60, 40) and the corresponding deadlines d(2, 3, 1, 4). find out max profit using greedy method.

Sol: Descending order of jobs where $n=4$



This is

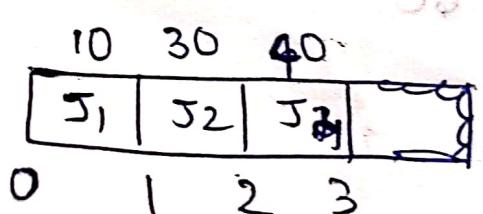
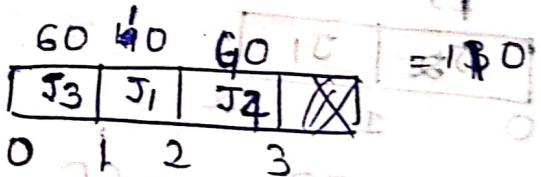
P(10, 30, 60, 40)

d(2, 3, 1, 4)

Descending order: (J₃, J₄, J₂, J₁)

(1, 4, 3, 2)

Gantt chart:



$$= 180$$

et	et		
0	1	2	3

$$\begin{array}{|c|c|} \hline 60 & 30 \\ \hline J_1 & J_2 \\ \hline \end{array} = 90$$

we can change the sequence corresponding to max profit

3. Find out greedy method of profit

profit weight

Sol: Desc profit

Gantt chart

\therefore we can

the seq

corresponding

Q. Find greedy profits

60	30	40
J ₃	J ₂	J ₄

$$= 130$$

we can get maximum profit by selecting the sequence of jobs (J₃, J₂, J₄) by corresponding deadlines.
max profit obtained = 130

Q. Find out the maximum profit by using the greedy method. the no. of jobs are 5. corresponding profits and weights are given below.

	J ₁	J ₂	J ₃	J ₄	J ₅
profit	2	4	3	1	10
weight	3	3	3	4	4

Sol: Descending order of jobs based on profits.

(J₅, J₂, J₃, J₁, J₄)

(10, 4, 3, 2, 1)

Gantt chart (size of maximum deadline)

J ₅	J ₁	J ₃	J ₂	
0	1	2	3	4

2	3	4	5	6	7	8
J ₁	J ₃	J ₂	J ₅			

∴ we can get the maximum profit by selecting the sequence of jobs (J₁, J₃, J₂, J₅) by corresponding deadlines.

max profit obtained = 19.50

Q. Find out the maximum profit by using the greedy method. The no. of jobs corresponding profits & weights are given below.

0.0 + 0.1 = fitting minimum
0.81 = fitting maximum

	J1	J2	J3	J4	J5	J6	J7	J8	J9
Profit	15	20	30	18	18	10	23	16	25
Weight	7	2	5	3	4	5	2	3	3

Sol: Descending order of jobs based on profit

$$(J_3, J_9, J_7, J_2, J_4, J_5, J_8, J_1, J_6)$$

$$(30, 25, 23, 20, 18, 18, 16, 15, 10)$$

Gantt chart

20	23	25	18	30	15	16
J2	J4	J9	J5	J3	J1	J8

$$maximum\ profit = 20 + 23 + 25 + 18 + 30 + 15 + 16 = 147$$

\therefore we can get maximum profit by selecting

the sequence of jobs $(J_2, J_4, J_9, J_5, J_3, J_1, J_8)$

by corresponding deadlines.

$$maximum\ profit\ obtained = 147$$

Q. Find out the maximum profit by using greedy method. All the jobs are given below along with deadline and profits.

Job	J1	J2	J3	J4	J5
Profit	60	100	20	40	20
Deadline	2	1	3	2	1

Sol: Descending order of jobs based on profit

$$(J_2, J_1, J_4, J_3, J_5)$$

Gantt chart

J2	2J1	J3
0	1	2

$$maximum\ profit = 100 + 60 + 20 \\ = 180$$

\therefore we can get the sequence corresponding to maximum profit

Q. Find out the greedy method along with

Jobs	Deadline	Profit
J2	1	100
J1	2	60
J4	3	40
J3	4	20

Sol: Descending profit:

J

Gantt chart

0 1 2 3

maximum

200

\therefore we can get the sequence corresponding to maximum profit

Knap Sac

Q. Find the n

problem, object/item, weights & p

w

~~based on profit:~~

~~J₂, J₃, J₁~~

~~maximum profit obtained = 180.~~

~~we can get maximum profit by selecting the sequence of jobs (J₂, J₁, J₃) by corresponding deadlines.~~

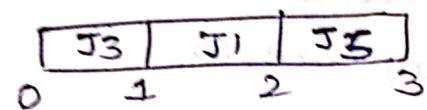
Q. Find out maximum profit by using greedy method. All the jobs are given below along with deadline and profits.

Jobs	J ₁	J ₂	J ₃	J ₄	J ₅
Deadline	2	1	2	1	3
Profit	100	19	27	25	15

Sol: Descending order of Jobs based on profit:

J₁, J₃, J₄, J₂, J₅

Gantt chart:



$$\text{maximum profit} = 27 + 100 + 19 \\ \text{maximum profit} = 146$$

\therefore we can get maximum profit by selecting the sequence of jobs (J₃, J₁, J₂) by corresponding deadlines.

maximum profit obtained = 146

fractional KSP (greedy method)

Knap Sack Problem < 0/1KSP (dynamic programming)

Q. Find the maximum profit by using knapsack problem, where $M=20, n=3$. The corresponding object/item weights & profits are given as

	item1	item2	item3	
P	25	24	15	
W	18	15	10	

m-knapsack capacity
n-no. of items

Sol: Greedy about weight: Knapsack

Knapsack weight $m = 20$
Greedy about weight. In knapsack
so, greedy about weight.
1st we can store place item 3. The item size is
10, item 3's weight is 10, item 3's profit is 15
 $20 - 10 = 10$

Now, the remaining bag size is $20 - 10 = 10$.
Next, we can place item 2. The item size is
15 but, the capacity of knapsack is only 10.

maximum profit = 31
 $\frac{31}{28}$

maximum profit = 31
 $\frac{31}{28}$

Greedy about profit: Knapsack

1st we can place item 3. It's size is 18, and
now, remaining bag size is $20 - 18 = 2$

Next, we can place item 2. It's
size is 24, but the capacity of
knapsack is only 24. item 2's size is 24 and
maximum profit is $28 \cdot 2 + 0$ since size is 24

$\frac{28 \cdot 2}{28} = 56$

item 2's size is 24 and
 $\frac{28 \cdot 2}{28} = 56$

item 2's size is 24 and
 $\frac{28 \cdot 2}{28} = 56$

item 2's size is 24 and
 $\frac{28 \cdot 2}{28} = 56$

item 2's size is 24 and
 $\frac{28 \cdot 2}{28} = 56$

item 2's size is 24 and
 $\frac{28 \cdot 2}{28} = 56$

item 2's size is 24 and
 $\frac{28 \cdot 2}{28} = 56$

W	P
15	24
5	7.5
5	7.5
31.5	

$$\frac{15}{20} = 0.75$$

$$\frac{7.5}{20} = 0.375$$

$$\frac{7.5}{20} = 0.375$$

$$\frac{31.5}{20} = 1.575$$

1st we store item 2.

The item size is 15.
Now, remaining bagsize
 $m = 20 - 15 = 5$.

Next, we can place item 3.

The item size is 10 but, the
capacity of knapsack is only 5.

$$\text{maximum profit} = 31.5$$

Q. Find out the maximum profit by using
fractional knapsack problem (Greedy method), where
 $m=15$, $n=7$. The profit and weights of each
item is given below.

Given data,
The capacity of
knapsack $m=15$.

The no. of items
 $n=7$

obj	1	2	3	4	5	6	7
P	10	5	15	7	6	18	3
W	2	3	5	7	1	4	1

Step 1: Now, calculate the $\frac{P}{W}$ ratio.

Plw ratio:	obj	1	2	3	4	5	6	7
P	10	5	15	7	6	18	3	
W	2	3	5	7	1	4	1	
Plw	5	1.66	3	1	6	4.5	3	

Des Order
of items based
on Plw's

5, 1, 6, 3, 7, 2,

4

1st we store item 5. The item size is

14. Now, remaining bagsize is 14

Next, we can place item 1. It's size

is 2. Now, remaining bagsize

is 12. Next, we can place item 6. It's size

is 4. Now, remaining bag size is 8. Place item 3. It's size

is 5. Now, place item 7.

Item 1	2
Item 6	4

Item 3	5
Item 7	1

Knapsack	
W	15-4

$$4 \quad P \quad 15-4 \\ 4 \quad 11 \quad = 11$$

$$2 \quad 10 \quad 11-2=9$$

$$4 \quad 18 \quad 12-4=8$$

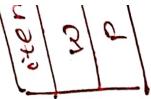
$$3 \quad 15 \quad 8-5=3$$

$$1 \quad 8 \quad 5-1=4$$

$$2 \quad 3.33$$

$$\text{Max Profit} = 55.33$$

Ex: Find a maximum profit by using greedy knapsack method and each and every problem whose NP-HC is given below.



Sol: Given data:
The capacity of knapsack, $m = 15$
The capacity of knapsack, $P = 5$

The no. of units, $n = 5$
Step 1: Find profit ratio and sort objects in descending order.

Obj	1	2	3	4	5
P	2	28	25	18	9
W	2	284	5	3	3
PLW	0.00	0.00	0.00	0.00	0.00

Step 2: Descending order of objects based on $\frac{P}{W}$ ratio.
2, 4, 3, 5, 1

Step 3: 1st place Obj 2. It's weight is 4
Now, remaining size is 11.

Place object 4. whose weight is 3
So, remaining size is 8

Now, place Obj 3. It's wt is 5
So, remaining size is 3 & Obj 1's $\frac{P}{W}$ is 28

Now, place Obj 5. It's size is 3
So, remaining size is 0 & Obj 1's $\frac{P}{W}$ is 28

Now, place Obj 1. Its $\frac{P}{W}$ is 28
So, remaining size is 0 & Obj 1's $\frac{P}{W}$ is 28

Now, place Obj 2. Its $\frac{P}{W}$ is 28
So, remaining size is 0 & Obj 2's $\frac{P}{W}$ is 28

Now, place Obj 3. Its $\frac{P}{W}$ is 28
So, remaining size is 0 & Obj 3's $\frac{P}{W}$ is 28

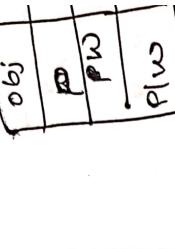
Now, place Obj 4. Its $\frac{P}{W}$ is 28
So, remaining size is 0 & Obj 4's $\frac{P}{W}$ is 28

Now, place Obj 5. Its $\frac{P}{W}$ is 28
So, remaining size is 0 & Obj 5's $\frac{P}{W}$ is 28

Sol: Given data:
Capacity

No. of units

Step 1: Given data:
Capacity



Step 2: Descr.

Step 3:

Step 4:

Step 5:

Step 6:

Step 7:

Step 8:

Step 9:

Step 10:

Step 11:

Step 12:

Step 13:

Step 14:

Step 15:

Step 16:

Step 17:

Step 18:

Step 19:

Step 20:

Step 21:

Step 22:

Step 23:

Step 24:

Step 25:

Step 26:

Step 27:

Step 28:

Step 29:

Step 30:

Step 31:

Step 32:

Step 33:

Step 34:

Step 35:

Step 36:

Step 37:

Step 38:

Step 39:

Step 40:

Step 41:

Step 42:

Step 43:

Step 44:

Step 45:

Step 46:

Step 47:

Step 48:

Step 49:

Step 50:

Step 51:

Step 52:

Step 53:

Step 54:

Step 55:

Step 56:

Step 57:

Step 58:

Step 59:

Step 60:

Step 61:

Step 62:

Step 63:

Step 64:

Step 65:

Step 66:

Step 67:

Step 68:

Step 69:

Step 70:

Step 71:

Step 72:

Step 73:

Step 74:

Step 75:

Step 76:

Step 77:

Step 78:

Step 79:

Step 80:

Step 81:

Step 82:

Step 83:

Step 84:

Step 85:

Step 86:

Step 87:

Step 88:

Step 89:

Step 90:

Step 91:

Step 92:

Step 93:

Step 94:

Step 95:

Step 96:

Step 97:

Step 98:

Step 99:

Step 100:

Step 101:

Step 102:

Step 103:

Step 104:

Step 105:

Step 106:

Step 107:

Step 108:

Step 109:

Step 110:

Step 111:

Step 112:

Step 113:

Step 114:

Step 115:

Step 116:

Step 117:

Step 118:

Step 119:

Step 120:

Step 121:

Step 122:

Step 123:

Step 124:

Step 125:

Step 126:

Step 127:

Step 128:

Step 129:

Step 130:

Step 131:

Step 132:

Step 133:

Step 134:

Step 135:

Step 136:

Step 137:

Step 138:

Step 139:

Step 140:

Step 141:

Step 142:

Step 143:

Step 144:

Step 145:

Step 146:

Step 147:

Step 148:

Step 149:

Step 150:

Step 151:

Step 152:

Step 153:

Step 154:

Step 155:

Step 156:

Step 157:

Step 158:

Step 159:

Step 160:

Step 161:

Step 162:

Step 163:

Step 164:

Step 165:

Step 166:

Step 167:

Step 168:

Step 169:

Step 170:

Step 171:

Step 172:

Step 173:

Step 174:

Step 175:

Step 176:

Step 177:

Step 178:

Step 179:

Step 180:

Step 181:

Step 182:

Step 183:

Step 184:

Step 185:

Step 186:

Step 187:

Step 188:

Step 189:

Step 190:

Step 191:

Step 192:

Step 193:

Step 194:

Step 195:

Step 196:

Step 197:

Step 198:

Step 199:

Step 200:

Step 201:

Step 202:

Step 203:

Step 204:

Step 205:

Step 206:

Step 207:

Step 208:

Step 209:

Step 210:

Step 211:

Step 212:

Step 213:

Step 214:

Step 215:

Step 216:

Step 217:

Step 218:

Step 219:

Step 220:

Step 221:

Step 222:

Step 223:

Step 224:

Step 225:

Step 226:

Step 227:

Step 228:

Step 229:

Step 230:

Step 231:

Step 232:

Step 233:

Step 234:

Q. Find a max profit by using greedy knapsack problem where $m=60$, $n=5$ and object profit & wt is given

item	1	2	3	4	5
w	5	10	15	22	12.5
P	30	40	45	72	90

Sol: Given data,
Capacity of knapsack, $m=60$
no. of units, $n=5$ objects to opt.

obj	1	2	3	4	5
P	30	40	45	72	90
P/w	5	10	15	22	12.5

Step1: P/w ratio
Obj 1: 5
Obj 2: 10
Obj 3: 15
Obj 4: 22
Obj 5: 12.5
Step2: Descending order of items based on P/w ratio.

Step3:
1, 2, 3, 4, 5

1st place obj 1, its weight is 5. Now
remaining wt is 55. Now

place, obj 2, its wt is 10.
Remaining wt is 45. Now, item 3 size

place obj 3, its wt is 15. Now, item 4 size

remaining wt is 20. So item 5 will fit.
Now, place obj 4. So item 5 will fit.

Maximum profit is 130. So item 5 will fit.

Maximum profit is 130. So item 5 will fit.

Maximum profit is 130. So item 5 will fit.

Maximum profit is 130. So item 5 will fit.

Maximum profit is 130. So item 5 will fit.



Q. Find the max profit using greedy knapsack problem, where $m=15$, $n=7$ and each object weight & profit are given below:

Obj	1	2	3	4	5	6	7
P	5	10	15	7	8	9	4
W	1	3	5	4	1	3	2

Sol: Given data,

Knapsack size, $m=15$
no. of objects = 7

Step 1: find profit/weight ratio

Obj	1	2	3	4	5	6	7
P	5	10	15	7	8	9	4
W	1	3	5	4	1	3	2
P/W	5	3.33	3	1.75	8	3	2

Step 2: Descending Order of Objects Based on P/W ratio

Obj 5, 1, 2, 3, 6, 7, 4

Step 3:

1st place obj 5. It's weight is 1

Remaining size is 14

Next place obj 1, It's weight is 4.

Remaining size is 13. Now

Place obj 2, It's weight is 3

Now, remaining size is 10

Now, place obj 3. Remaining

size is 5. Now, place obj 6

It's wt is 13 Remaining size is 9

Now, place 7. It's wt is 2.

Max Profit = 51

Obj 5	1
Obj 3	5
Obj 2	3
Obj 1	1
Obj 6	2

Obj 5	10	10
3	5	15
6	8	9
7	0	4

Q. Find the m knapsack problem each & every ob

Item

Sol: Given d knapsa no. of

Step 1: find p

Item	A
P	28
W	4
P/W	7

Step 2: descer

step3: 1st p

Remaining

Now, place

Remaining

Now, pla

but we

Maximum



Q. Find the max profit to using fractional knapsack problem, where $m=60$, $n=4$ where each & every object's wt & P are given below.

item	A	B	C	D
P	280	100	120	120
w	40	10	20	24
	7	10	6	5

Sol. Given data,
knapack size, $m=60$
no. of items = 4

Step 1: find P/w ratio

Item	A	B	C	D
P	280	100	120	120
w	40	10	20	24
	7	10	6	5

Step 2: descending order of items based on P/w ratio.

B, A, C, D

Step 3: 1st place item B. It's wt is 10.

Remaining w is 50.

Now, place item A. It's wt is 40.

Remaining wt is 10.

Now, place item C. It's wt is 20.

but we place only upto 10.

Maximum profit = $100 + 280 + 60$

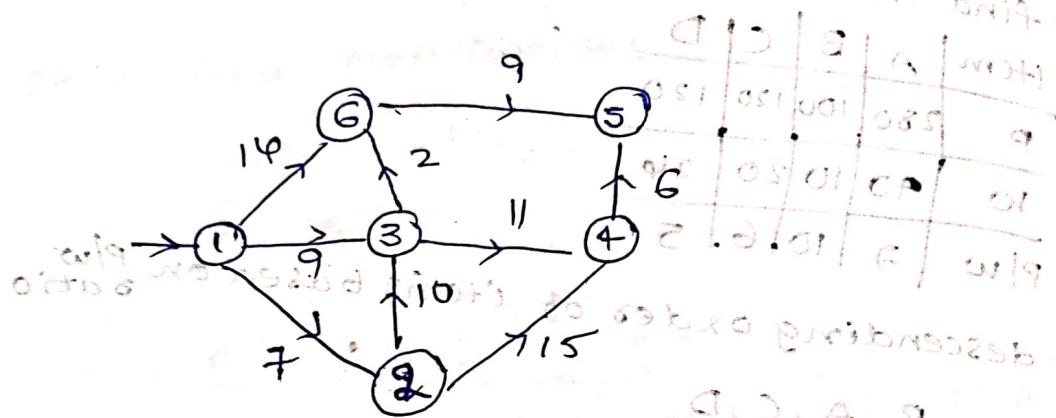
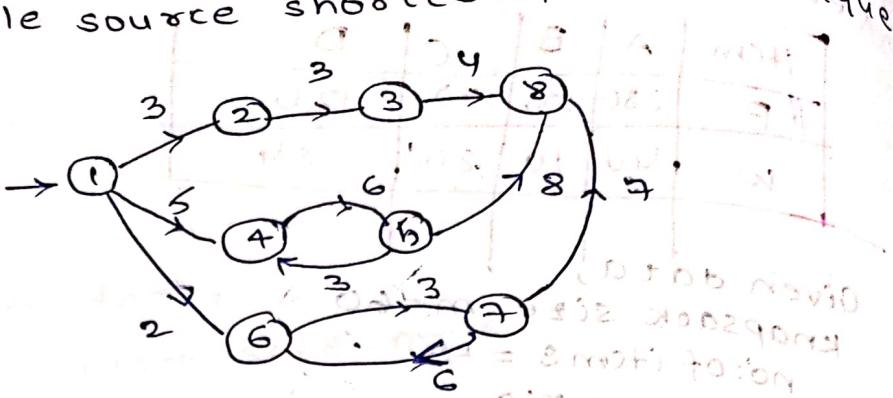
Item A	40
Item B	40
	60

item w	P
B 50	100
A 10	280
C 20	60
	460

2nd step
3rd step

Single Source Shortest Path

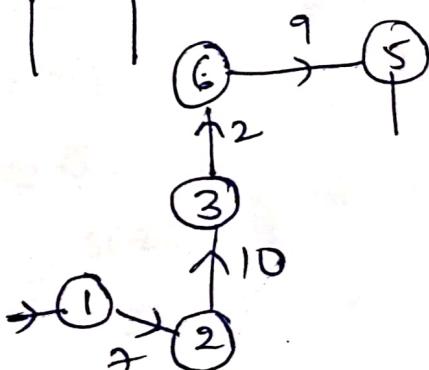
Eg: Find out the shortest path by using single source shortest path technique



Source	2	3	4	5	6
1	00	∞	∞	∞	00
2	7	9	∞	∞	16
3	10	45	00	00	00
4	17	22	00	00	00
5	20	26	00	00	19
6	00	28	00	00	00
7	34	00	00	00	00

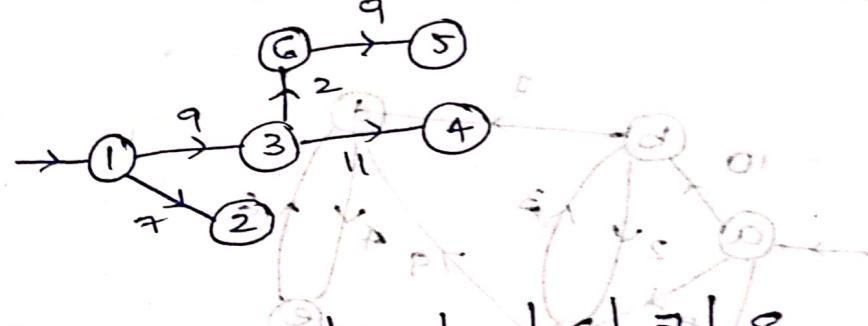
1, 2, 3, 6, 5

1, 2, 3, 6, 5, 4



source	2	3	4	5	6
1	7	9	∞	∞	14
1, 2		9	22	∞	14
1, 2, 3			20	∞	11
1, 2, 3, 6			20	20	

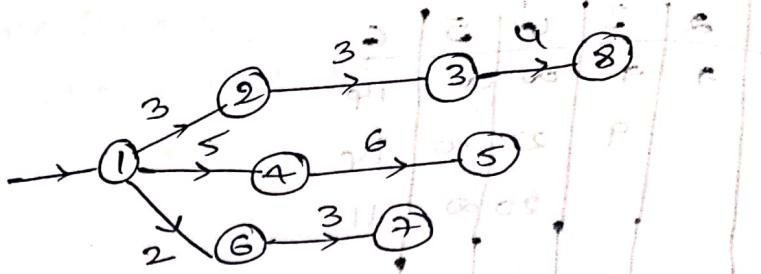
1, 2, 3, 6, 4, 5
1, 2, 3, 6, 4, 5, 8
1, 2, 3, 6, 4, 5, 8, 9
1, 2, 3, 6, 4, 5, 8, 9, 10
1, 2, 3, 6, 4, 5, 8, 9, 10, 11
1, 2, 3, 6, 4, 5, 8, 9, 10, 11, 12



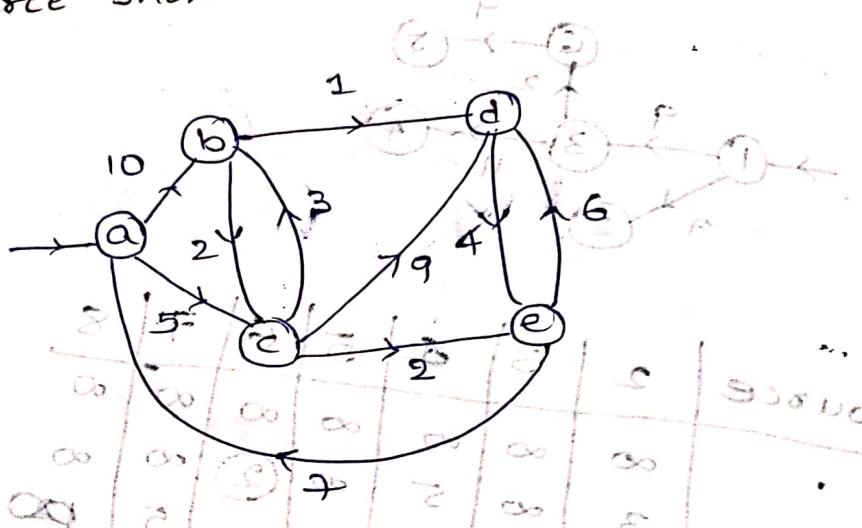
(Q1)

source	2	3	4	5	6	7	8
.	∞						
1	3	∞	5	∞	2	∞	5
1, 6	3	∞	5	∞	3	5	∞
1, 6, 2	6	∞	5	∞	3	∞	8
1, 6, 2, 4	6	∞	5	11	2	∞	12
1, 6, 2, 4, 7	6	∞	5	11	2	8	10
1, 6, 2, 4, 7, 3	P	∞	5	11	3	8	10
1, 6, 2, 4, 7, 3, 8	P	∞	5	11	3	8	10
1, 6, 2, 4, 7, 3, 8, 5	6	∞	5	11	3	8	10

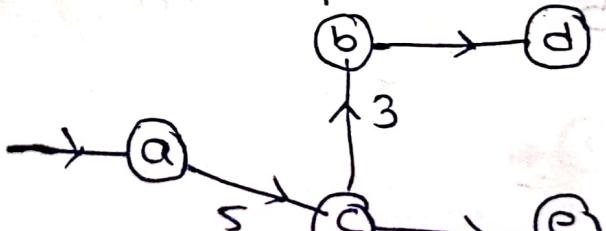




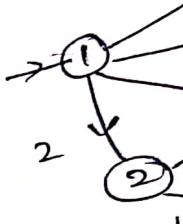
Q. Find out the shortest path by using single source shortest path technique.



source	b	c	d	e
a	∞	∞	∞	∞
a,c	8	-	14	∞
a,c,e	(1)	(8)	-	-
a,c,e,b			(9)	
a,c,e,b,d				



Q. Find out source



source

1

1, 2

1, 2, 5

1, 2, 5, 4

1, 2, 5, 6

1, 2, 5, 6,

draw graph

1, 2, 4, 5, 6

1, 2, 4, 5, 6

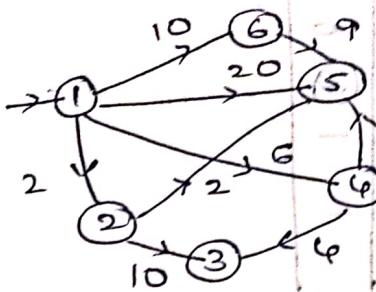
1, 2, 4, 5, 6

Q. Find source

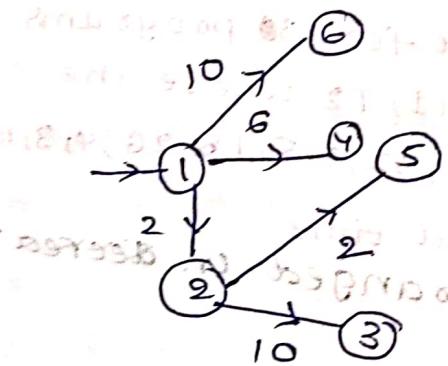
2

2

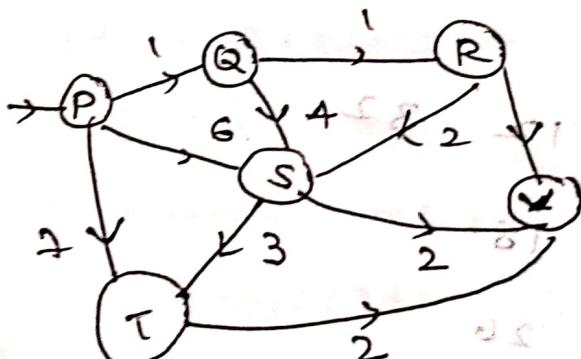
B. Find out the shortest-path using single source shortest path technique.



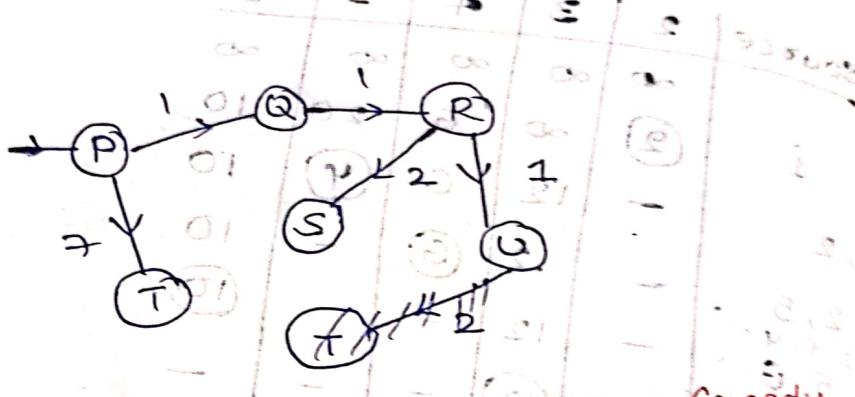
source	2	3	4	5	6
1	∞	∞	∞	∞	∞
1, 2	(2)	∞	(6)	(20)	10
1, 2, 5	-	12	6	4	10
1, 2, 5, 4	-	12	6	5	10
1, 2, 5, 6	-	12	6	5	10
1, 2, 5, 6, 3	-	12	-	-	-



B. Find out the shortest path using single source shortest path technique.



source	Q	R	S	T	U	V
P, Q, R, S, T	1	∞	6	7	∞	
P, Q, R, U	-	2	5	7	∞	
P, Q, R, V	-	-	4	7	3	
P, Q, R, S, U	-	-	5	5	-	
P, Q, R, S, T	-	-	-	3	-	
P, Q, R, S, U, V	-	-	-	3	2	
P, Q, R, S, T, V	-	-	-	3	2	



Optimal Storage on Tapes using Greedy method.

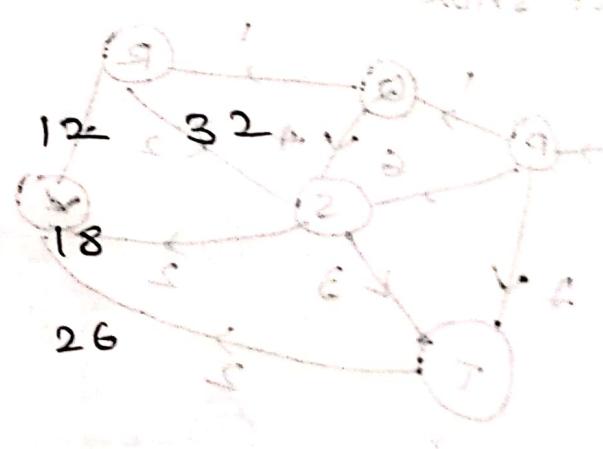
Q. Find the optimal placement for 13 programs on 3 tapes those are T_0, T_1, T_2 where the program lengths are 12, 5, 8, 32, 7, 5, 18, 26, 43, 11, 10, 6. Find the optimal retrieval time.

Step1: All the files are arranged in ascending order.

3, 4, 5, 6, 7, 8, 10, 11, 12, 18, 26, 32

Step2:

T_0	3	5	8	12	32
T_1	4	6	10	18	
T_2	7	11	26		



Step 3: Optimal retrieval time for tapes

$$T_0 = \frac{3 + (3+5) + (3+5+8) + (3+5+8+12) + (3+5+8+12+32)}{(3+5+8+12+32)} \\ = \frac{3+8+16+28+60}{(3+5+8+12+32)} \\ = 115$$
$$T_1 = 4 + (4+6) + (4+6+10) + (4+6+10+8) \\ = 4+10+20+38$$

$$T_2 = 5 + (5+7) + (5+7+11) + (5+7+11+26) \\ = 5+12+23+49$$

$$= 89$$

Average retrieval time is

$$\frac{T_0 + T_1 + T_2}{3} = \frac{276}{3} = 92$$

B. Find the optimal retrieval time where lengths of each files are 10, 20, 45, n=10 and lengths of each files are 10, 20, 45, 54, 23, 67. There are 3 tapes.

Step 1: Ascending order of all files:

$$10, 20, 45, 54, 23, 67$$

Step 2: Let us consider tapes as

$$T_0 \quad 1 \quad 10 \quad 45 \quad 70$$

$$T_1 \quad 3 \quad 20 \quad 54$$

$$T_2 \quad 7 \quad 23 \quad 67$$

Step 3: finding optimal retrieval time for T₀.

$$\Rightarrow 1 + (1+10) + (1+10+45) + (1+10+45+70)$$

$$= 1+11+56+126 = 194$$

finding retrieval time for T₁.

$$\Rightarrow 3 + (3+20) + (3+20+54)$$

$$= 3 + 103 = 106$$

finding retrieval time for T₂.

$$\Rightarrow 7 + (7+23) + (7+23+67) = 134$$

Step 4: Average retrieval time = $\frac{194 + 103 + 134}{3} = \frac{431}{3} = 143.33$

Q. Find the optimal retrieval time where $n=13$,
there are 3 tapes and length of 12, 34, 56, 73, 24, 11,
34, 56, 78, 91, 34, 34, 45, 56, 73, 78, 91, 91.

Sol: Step 1: Arrange lengths of files in ascending order.

Step 2: Let us consider tapes as T_1, T_2, T_3

	T_0	T_1	T_2	P_0	P_1	P_2
	11	34	45	73	91	
	12	34	56	78		
	24	34	56	91		

Step 3: Finding retrieval time for T_0

$$T_0 \Rightarrow 11 + (11+34) + (11+34+45) + (11+34+45+73) + (11+34+45+73+91)$$

$$\Rightarrow 11 + 45 + 90 + 163 + 254$$

$$\Rightarrow 563$$

$$T_1 \Rightarrow 12 + (12+34) + (12+34+56) + (12+34+56+78)$$

$$\Rightarrow 12 + 46 + 102 + 180$$

$$\Rightarrow 340$$

$$T_2 \Rightarrow 24 + (24+34) + (24+34+56) + (24+34+56+91)$$

$$(24+34+56+91) + 1 = 24 + 58 + 114 + 208 + 1 = 401$$

Step 4: Average retrieval time = $\frac{563 + 340 + 401}{3} = \frac{1304}{3} = 434.6$

$$\frac{563 + 340 + 401}{3} = \frac{1304}{3} = 434.6$$

$$401 = (f_0 + f_1 + f_2) + (e_0 + e_1 + e_2) + r =$$

Optimal
Find out to
merge for
do step 1:
2, 3, 5,
merge 15
It is in
the 10,
It is no
Ascending
1, a
 $\Rightarrow 10$
Arrang'g
merg
Arrang'

Optimal Merge Pattern Using Greedy method

Q. Find out total no. of merges (or) no. of optimal merge for the given set of files 2, 9, 7, 5, 3, 13
Arrange all files into ascending order.

Ans Step 1: Ascending order.

2, 3, 5, 7, 9, 13

merge 1st 2 files: 5, 5, 7, 9, 13.
It is in ascending order. So, merge 1st 2 files
are 10, 7, 9, 13.

It is not in ascending order. Arrange in
ascending order and merge 1st 2 files.
1, 9, 10, 13

\Rightarrow 16, 10, 13

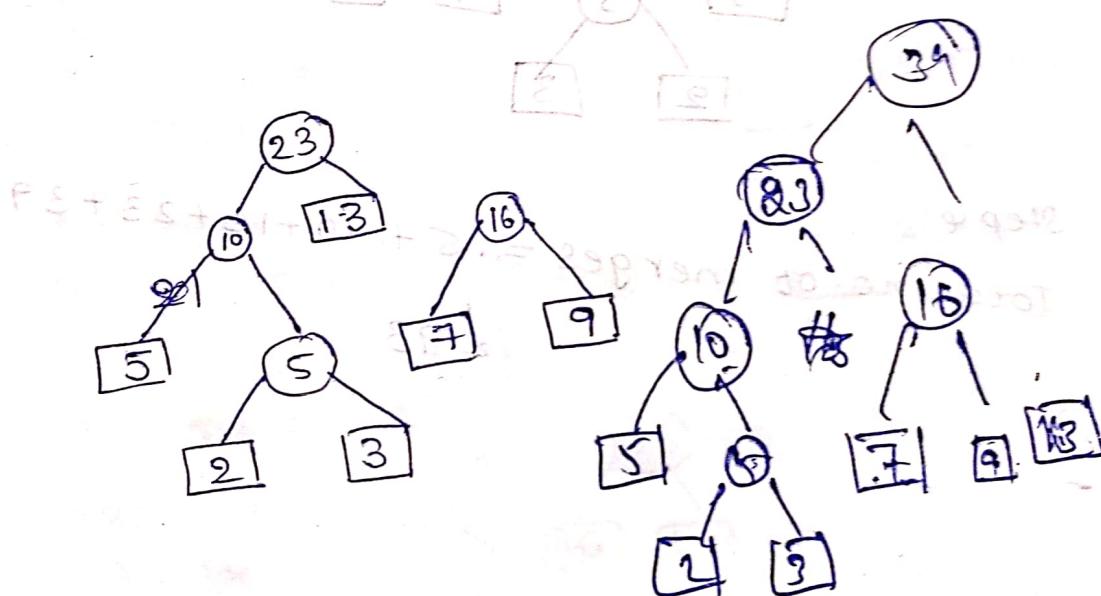
Arrange in Ascending order: 10, 13, 16.

merge 1st 2 files: 10, 13, 16.

merge 1st 2 files: 16, 23

Arrange in Ascending order: 16, 23

merge 1st 2 files: 39.



2, 3, 5, 7, 9, 13

5, 5, 7, 9, 13

10, 7, 9, 13

7, 9, 10, 13

16, 10, 13

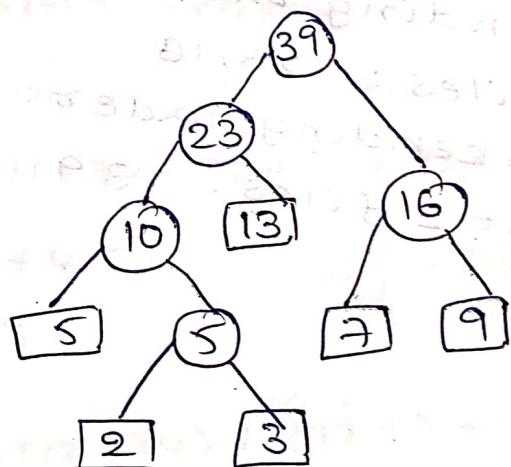
10, 13, 16

23, 16

16, 23

39

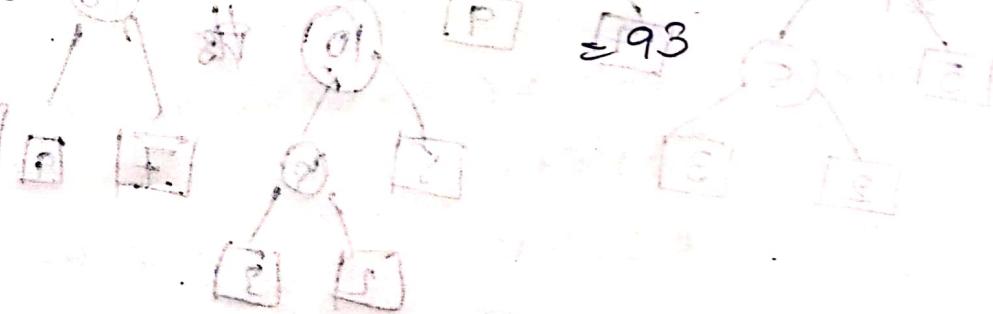
Step 3: Construction of tree.



Step 4:

$$\text{Total no. of merges} = 5 + 10 + 16 + 23 + 39$$

$$= 93$$



Step 5: Average merging

Time complexity of merge sort is $O(n \log n)$

B. Find out to set of files
sol: Step 1: A
order. 2, 5
step 2: mes
list is in A
otherwise
merge oper

let t
2, 5, 7, 11
7, 7, 11
16, 11

11, 14

28, 30

25

16

37

44, 56, 2

33, 55

33, 55

25, 30

30, 35

Q. Find out total no. of merges for the set of files. 2, 5, 7, 11, 16, 21, 13, 28, 42

Sol: Step1: Arrange the given files in ascending order. 2, 5, 7, 11, 16, 17, 21, 26, 28, 42

Step2: Merge 1st 2 files. And if the obtained list is in Ascending order, merge 1st 3 files otherwise arrange in Ascending order & perform merge operations. repeat until only 1 element is left.

left: 2, 5, 7, 11, 16, 17, 21, 26, 28, 42

└─ 7, 11, 16, 17, 21, 26, 28, 42

└─ 16, 11, 16, 17, 21, 26, 28, 42

└─ 11, 14, 16, 17, 21, 26, 28, 42

└─ 25, 16, 17, 21, 26, 28, 42

└─ 16, 17, 21, 25, 26, 28, 42

└─ 33, 21, 25, 26, 28, 42

└─ 21, 25, 26, 28, 33, 42

└─ 46, 26, 28, 33, 42

└─ 26, 28, 33, 42, 46

└─ 54, 33, 42, 46

└─ 33, 42, 46, 54

└─ 75, 46, 54

└─ 46, 54, 75

└─ 110, 75

└─ 75, 110

└─ 185

Step3: Represent in the form of tree

26, 33, 38, 42, 46

59, 38, 42, 46

38, 42, 46, 59

80, 46, 59

46, 59, 80

105, 80

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

80, 105

185

105

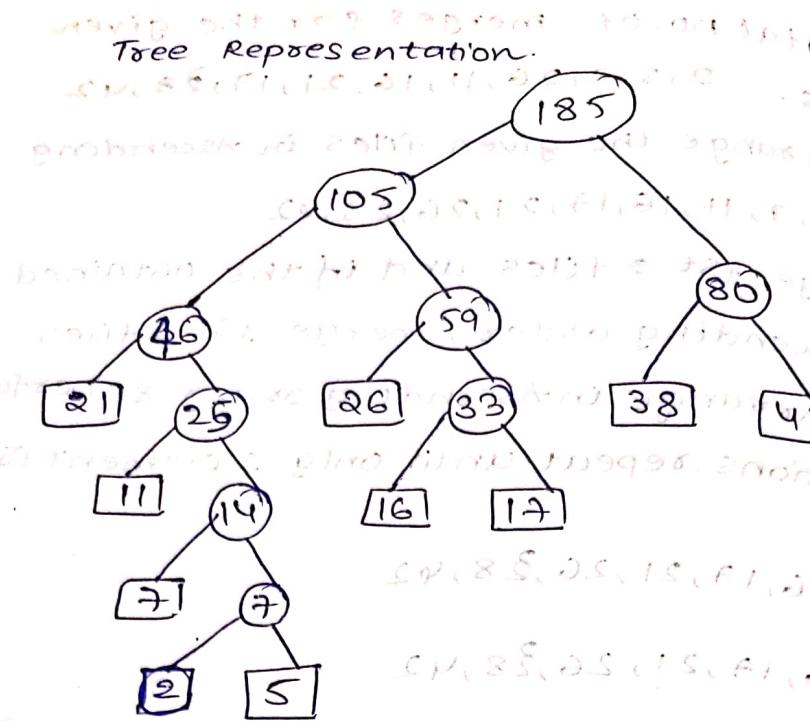
80, 105

185

105

80, 105

</div



Step 4: Total no. of merges = $\sum d_i x_i$

$$7 + 14 + 25 + 33 + 46 + 59 + 80 + 105 + 185$$

$$= 554$$

We verify no. of merges using $\sum d_i x_i$

$$6 \times 2 + 6 \times 5 + 5 \times 7 + 4 \times 11 + 21 \times 3 + 16 \times 4 + 17 \times 4 +$$

$$21 \times 3 + 3 \times 26 + 9 \times 38 + 2 \times 42$$

$$= 554$$

Q. construct optimal merge tree for the given files

8, 2, 9, 1, 12, 10, 18, 15, 14, 17

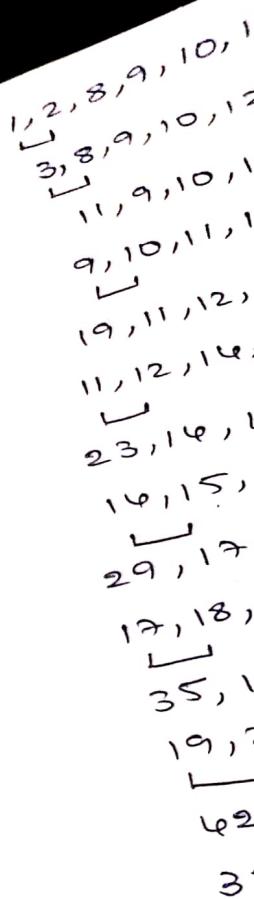
Sol: Step 1: Arrange them in Ascending order.

1, 2, 8, 9, 10, 12, 14, 15, 17, 18

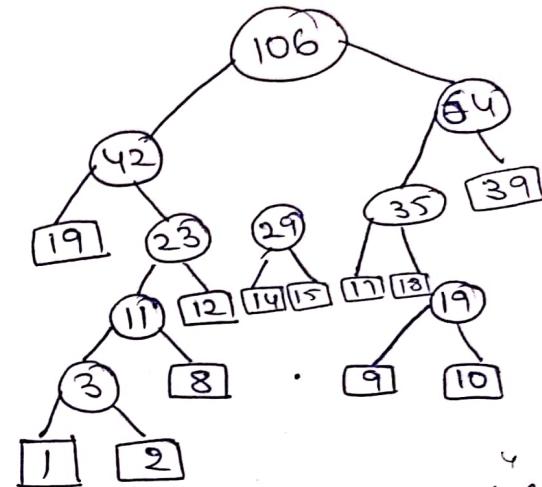
Step 2: If Ascending order, merge 1st two files

If not arrange them and merge. repeat until
only one node will left

Step 3: Tree



1, 2, 8, 9, 10, 12, 14, 15, 17, 18
3, 8, 9, 10, 12, 14, 15, 17, 18
11, 9, 10, 12, 14, 15, 17, 18
9, 10, 11, 12, 14, 15, 17, 18
19, 11, 12, 14, 15, 17, 18
11, 12, 14, 15, 17, 18, 19
23, 14, 15, 17, 18, 19
14, 15, 17, 18, 19, 23
29, 17, 18, 19, 23
17, 18, 19, 23, 29
35, 19, 23, 29
19, 23, 29, 35
42, 29, 35
35, 29, 42
64, 42
42, 64
106



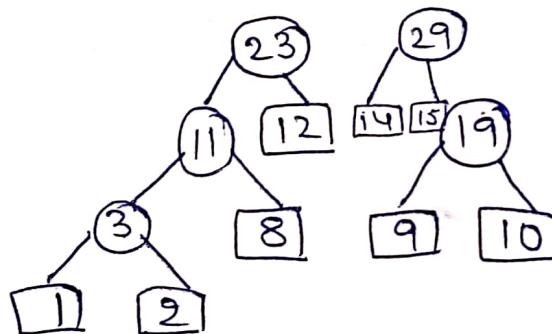
$$\begin{array}{r}
 & 4 \\
 10 & 6 \\
 - & 64 \\
 \hline
 & 42 \\
 & 23 \\
 & 29 \\
 & 35 \\
 & - \\
 & 14 \\
 & 19 \\
 \hline
 2 & 3 \\
 \hline
 3 & 3 & 2
 \end{array}$$

Step 3: Tree Representation.

Step 4:

Total no. of merges =

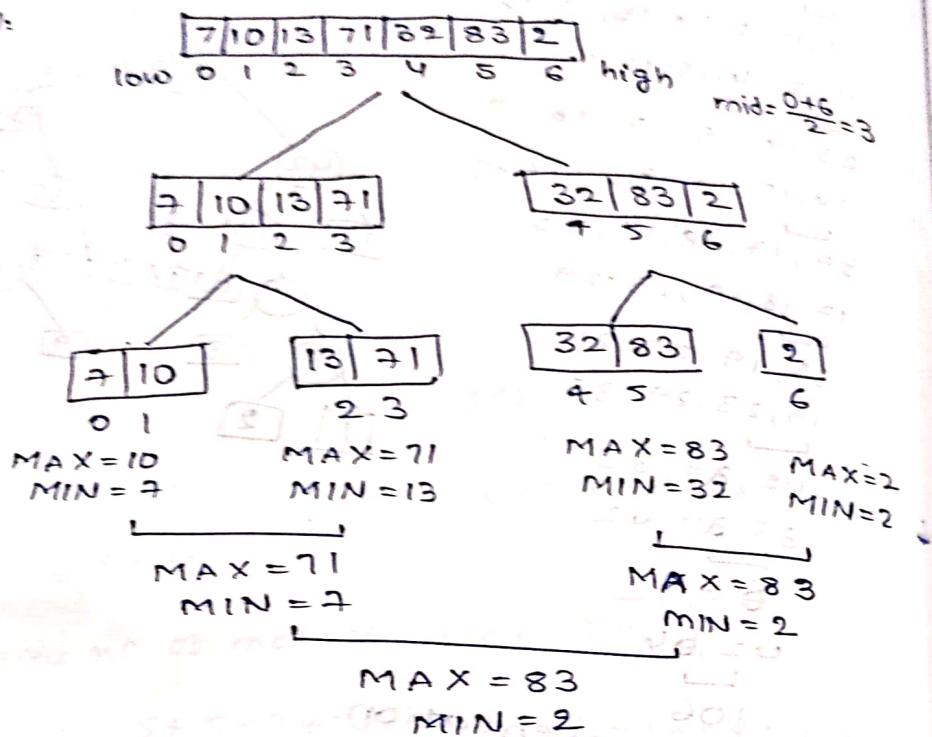
$$\begin{array}{r}
 3 + 11 + 19 + 23 + 29 + \\
 35 + 64 + 42 + \\
 106 \\
 = 332
 \end{array}$$



Finding a maximum & minimum element by using divide & conquer method.

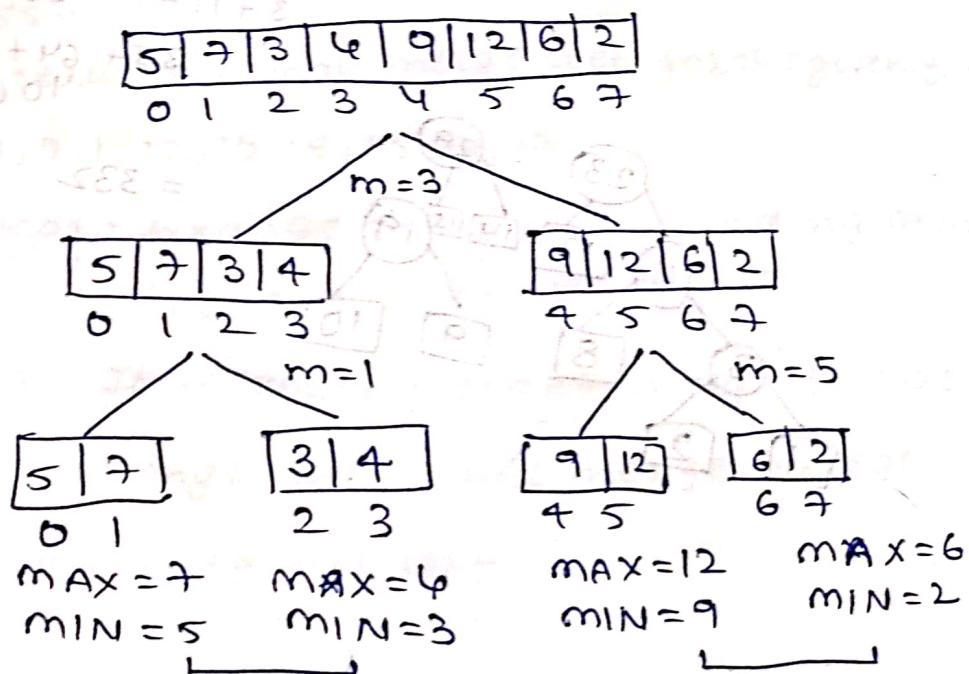
Q. Find a maximum and minimum element using the given array. The array size is 7. ($n=7$). The elements are 7, 10, 13, 71, 32, 83, 2.

Sol:



Q. The array size $n=8$, the list of elements are

5, 7, 3, 4, 9, 12, 6, 2.



a. find a maxim
the list. 6, 3
i.e. G 3
a 1

G | 3 | 4
0 1 2
 $m=$
G | P
0 7
MAX = 6
MIN = 3

MAX =
MIN =
(Algorithm)
n

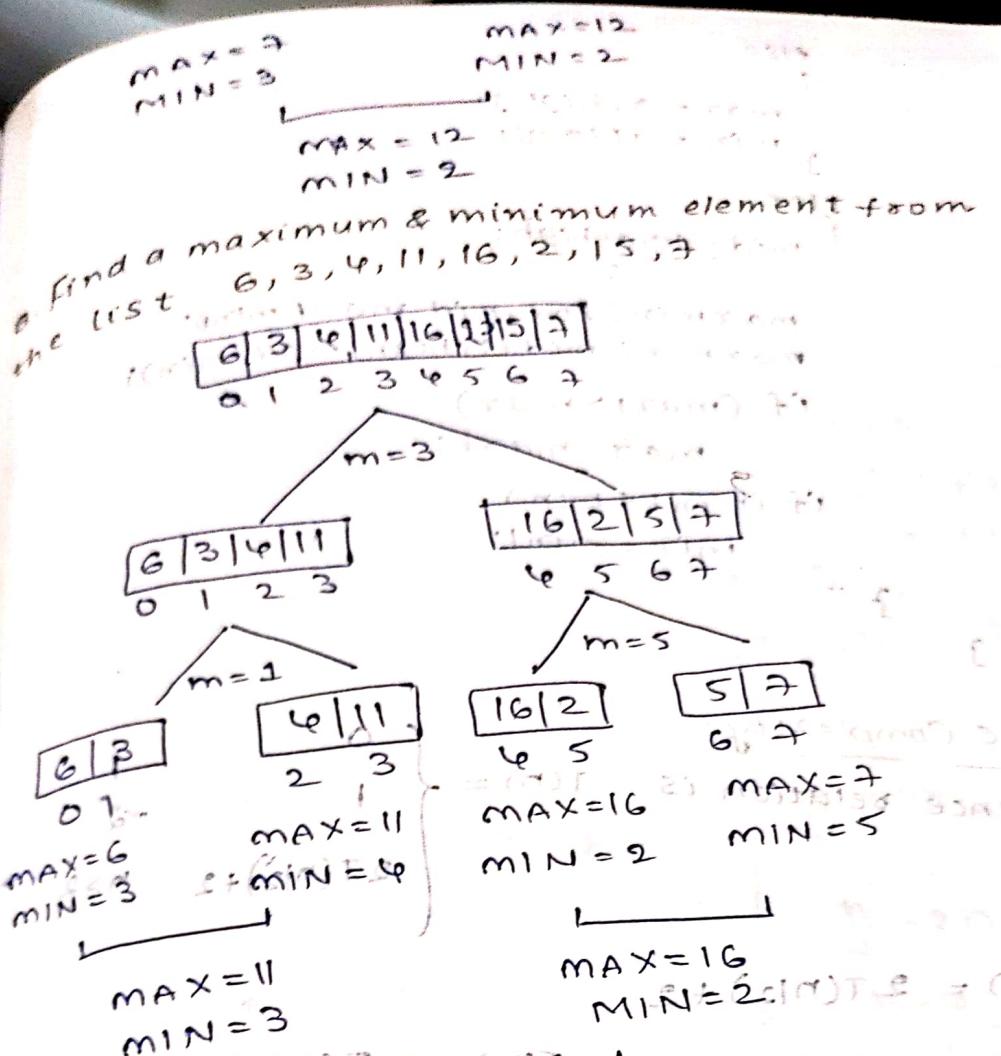
$s = \frac{n(n+1)}{2}$
Algorithm:

Algorithm

```

if (low < high)
{
    max
}
else if
{
    if
}

```



Algorithm:

Algorithm maxmin(low , $high$, max , min)

```

{ if ( $low == high$ )
  {
     $max = min = a[low]$ 
  }
  else if ( $low == high - 1$ )
  {
    if ( $a[low] > a[high]$ )
      {
         $max = a[low]$ ;
         $min = a[high]$ ;
      }
  }
}
  
```

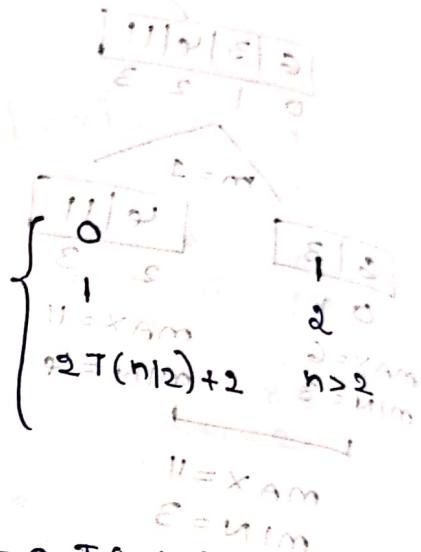
```

else
{
    max = a[high];
    min = a[low];
}
else
{
    mid =  $\frac{low+high}{2}$ ;
    maxmin(low, mid, max, min);
    maxmin(mid+1, high, max, min);
    if (max1 > max)
        max = max1;
    if (min1 < min)
        min = min1;
}
}
}

```

Time Complexity:

Recurrence relation is $T(n) = \Theta(n)$



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

using Master theorem: $T(n) = aT(n/b) + \Theta(n^k \log p)$

$$a=2, b=2, k=0, p \leq \Theta(n^2)$$

$$a > b^k$$

$$2 > 2^0$$

use 1st formula.

$$T(n) = \Theta(n \log_b a)$$

$$= \Theta(n \log_2 2)$$

$$= \Theta(n)$$

$$\log_p n = 2$$

$$\Rightarrow n^2 = p$$

Back-substitution

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

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

$$\vdots$$

$$\Rightarrow 2^k T(1)$$

$$\Rightarrow n/2$$

$$\Rightarrow n$$

using Back Substitution

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

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

$$= 4T(n/4)$$

$$T(n) = 4(2T(n/8))$$

$$= 8T(n/8)$$

Quick Sort:

e.g.: 2, 9, 7, 5, 3, 1
pivot: 2

2	9	7
5	3	1

Pivot

2	1
0	1

pivot

2

2

(2+5)=7
5+3=8
5+2=7
pivot

2

2
pivot

2

2
pivot

2

2
pivot

2

$$\begin{aligned}
 & n_{12}^k = 2 \\
 & \Rightarrow 2^k = n_{12} \\
 & n_{12} \cdot T(2^k/k) + 4\left(1 - \frac{2}{2^k}\right) \\
 & \Rightarrow \frac{n}{2} + 4 - \frac{8}{2^k} \\
 & \Rightarrow O(n)
 \end{aligned}$$

$$\begin{aligned}
 & \Rightarrow 2^k(T(n_{12}^k)) + \frac{\alpha(r_{n-1})}{\delta-1} \\
 & \quad \begin{array}{l} \delta=2 \\ \alpha=2 \\ n=k \end{array} \\
 & \quad + \frac{2(2^k-1)}{2-1} \\
 & \Rightarrow 2^{k+1}-2
 \end{aligned}$$

$$\Rightarrow 2^k(T(n_{12}^k)) + 2^{k+1}-2$$

$$\begin{aligned}
 & n_{12}^k = 2 \\
 & \Rightarrow 2^k = n_{12}
 \end{aligned}$$

$$\Rightarrow \frac{n}{2}(T(2^k/k)) + n-2$$

$$\frac{n}{2} + n - 2$$

$$\Rightarrow \frac{3n}{2} - 2 + (s/1^k)T(x_0)$$

$$s=0 \quad \Rightarrow \quad \frac{3n-4}{2} \quad O(n)$$

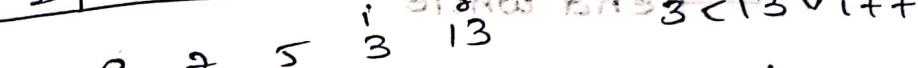
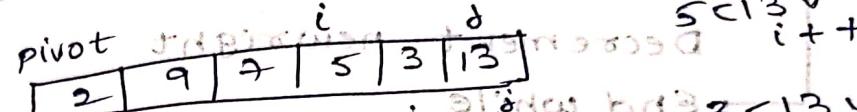
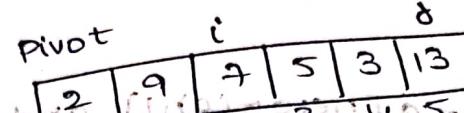
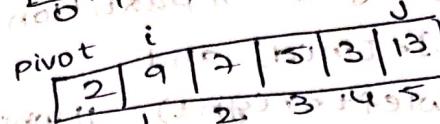
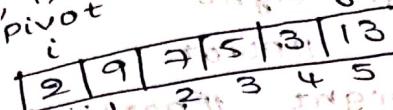
$$s=1 \quad \Rightarrow \quad \frac{3n-2}{2} + (x_0-1)N + (x_{s(N)})T(x_0)$$

$$s=2 \quad \Rightarrow \quad \frac{3n-4}{2} + (x_0-2)N + (x_{s(N)})T(x_0)$$

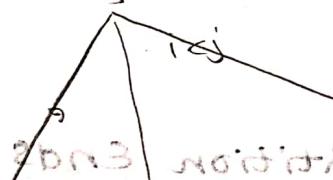
$$s=3 \quad \Rightarrow \quad \frac{3n-6}{2} + (x_0-3)N + (x_{s(N)})T(x_0)$$

Quick Sort:

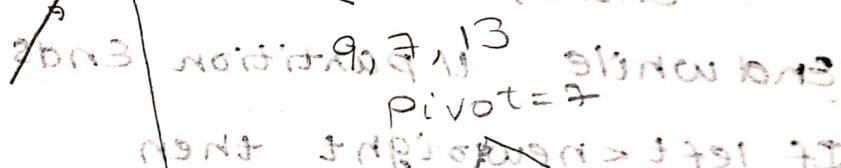
Eg: 2, 9, 7, 5, 3, 13.
pivot



Pivot = 5

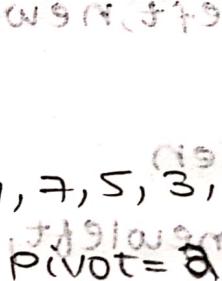


pivot = 7



Pivot = 2

2, 3



Pivot = 5

9, 7, 5, 3, 13

pivot = 9

7, 5, 3

Pivot = 7

= 2 /

Algorithm:

Quicksort(a, left, right):

 newleft = left, newright = right

 mid = a[(left+right)/2] // pivot

 while newleft ≤ newright // partition

 while (a[newleft] <= a[mid]) and

 newleft < right) // while loop₁

 Increment newleft

 end while

 while (a[mid] < a[newright]) and

 newright > left) // while loop₂

 Decrement newright

 end while

 if newleft ≤ newright, then

 swap(a[newleft], a[newright])

 Increment newleft and Decrement
 newright

 end if

end while // partition ends

if left < newright then

 call Quicksort(a, left, newright) // sort left
 Subarray

End If

if newleft < right then

 call Quicksort(a, newleft, right) // sort right
 Subarray

End If.

Time Complexity

$$T(n) = \begin{cases} 1 & n=1 \\ 2T(\frac{n}{2}) + 1 & n > 1 \end{cases}$$

Recurrence Relation

Time Complexity $T(n) = \begin{cases} 1 & n=1 \\ 2T(n/2) & n > 1 \end{cases}$

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

$$= 2 [2 T(n/4) + 1]$$

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

$$= 8 T(n/8) + 4$$

$$= 16 T(n/16) + 8$$

$$= 32 T(n/32) + 16$$

$$= 64 T(n/64) + 32$$

$$= 128 T(n/128) + 64$$

$$= 256 T(n/256) + 128$$

$$= 512 T(n/512) + 256$$

$$= 1024 T(n/1024) + 512$$

$$= 2048 T(n/2048) + 1024$$

Time Complexity

Average Case

Worst Case

$$Time\ Complexity \quad T(n) = \begin{cases} 1 & n=1 \\ 2T(n/2) + n & n > 1 \end{cases}$$

$$\begin{aligned}
 T(n) &= 2T(n/2) + n & ; n \geq 2 \\
 &= 2 \left[2T(n/4) + \frac{n}{2} \right] + n \\
 &= 4T(n/4) + n + n \\
 &= 4 \left[2T(n/8) + \frac{n}{4} \right] + n + n \\
 &= 8T(n/8) + n + n \\
 &= 8T(n/8) + 3n \\
 &= 2^k \cdot T(n/2^k) + kn \\
 n/2^k &= 1 \Rightarrow 2^k = n \\
 &\Rightarrow n + n \log_2 n
 \end{aligned}$$

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

Time Complexity is $O(n \log n)$

Average Case Time Complexity is $O(n \log n)$

Worst Case Time Complexity is $O(n^2)$

MergeSort Algorithm:

Algorithm MergeSort(left, right),

1. if(left < right)
2. mid = (left + right) / 2
3. mergesort(left, mid)
4. mergesort(mid, right)
5. mergeSort(left, mid, right)
6. End if

Algorithm Merge(left, mid, right):

1. i=j=left, k=mid+1
2. while j ≤ mid and i ≤ right
3. if a[i] < a[k]
4. Temp[i] = a[j];
5. Else
6. temp[i] = a[k];
7. End if
8. End while
9. while j ≤ mid
10. Temp[i] = a[j];
11. End while
12. for i=left to k
13. a[i] = temp[j]
14. End for

Time complexity for the mergesort is $n \log n$
 (Avg, best, worst)

UNIT-3

0-1 Knapsack prob

E. n=4, m=21, P =
 p and w are P
 item.

formula: S^{i+1}

Initially, $S^0 = \{0\}$

$S^{i+1} = S^i \cup S^i$

$S^1 = S^0 \cup S^0$

$S^2 = S^1 \cup S^1$

$S^3 = S^2 \cup S^2$

$S^4 = S^3 \cup S^3$

$S^0 = \{2^{10}\}$

$S_1^0 = \{0, 0\}$

$S^1 = \{2, 10\}$

$S^2 = \{0, 0\}$

$S^3 = \{(0, 0)\}$

$S^4 = \{(0, 0)\}$

$S^5 = \{(0, 0)\}$

$S^6 = \{(0, 0)\}$

$S^7 = \{(0, 0)\}$

$S^8 = \{(0, 0)\}$

$S^9 = \{(0, 0)\}$

$S^{10} = \{(0, 0)\}$

$S^{11} = \{(0, 0)\}$

$S^{12} = \{(0, 0)\}$

$S^{13} = \{(0, 0)\}$

$S^{14} = \{(0, 0)\}$

$S^{15} = \{(0, 0)\}$

$S^{16} = \{(0, 0)\}$

$S^{17} = \{(0, 0)\}$

$S^{18} = \{(0, 0)\}$

$S^{19} = \{(0, 0)\}$

$S^{20} = \{(0, 0)\}$

$S^{21} = \{(0, 0)\}$

UNIT-3

0-1 Knapsack Problem:

Given: $n=4, m=21, P = (2, 5, 8, 1), W = (10, 15, 6, 9)$
 P and W are profit & weights of each & every item.

$$\text{formula: } S^{i+1} = S^i \cup S_i^i$$

$$\text{initially, } S^0 = \{(0, 0)\}$$

$$S^{i+1} = S^i \cup S_i^i$$

$$S^1 = S^0 \cup S_1^0$$

$$S^2 = S^1 \cup S_1^1$$

$$S^3 = S^2 \cup S_1^2$$

$$S^4 = S^3 \cup S_1^3$$

$$S_1^0 = \{(2, 10)\}$$

$$S_1^0 = \{(0, 0) \oplus (2, 10) = (2, 10)\}$$

$$\rightarrow S^1 = \{(2, 10) \quad P_1 w_1 \quad P_2 w_2\}$$

$$\rightarrow S^1 = \{(0, 0) \quad (2, 10)\}$$

$$S_1^1 = S^1 \cup S_1^1$$

$$S_1^1 = \{(0, 0), (2, 10), (5, 15)\}$$

$$= \{(5, 15), (7, 25)\}$$

$$S^2 = \{(0, 0), (2, 10), (5, 15), (7, 25)\}$$

$$P_1 < P_2 \\ w_1 > w_2$$

$$\rightarrow S^2 = \{(0, 0), (2, 10), (5, 15)\}$$

$$S^3 = S^2 \cup S_1^2$$

$$S_1^2 = \{(0, 0), (2, 10), (5, 15), (8, 6)\}$$

$$= \{(8, 6), (10, 16), (13, 21)\}$$

$$S^3 = \{(0, 0), (2, 10), (5, 15), (8, 6), (10, 16), (13, 21)\}$$

$$\rightarrow S^3 = \{(0, 0), (8, 6), (10, 16), (13, 21)\}$$

$$S^4 = S^3 \cup S_1^3$$

$$S_1^3 = \{(0, 0), (8, 6), (10, 16), (13, 21), (1, 9)\}$$

2/2/23

the first time, I was very
excited because I had never
done it before. I was also
nervous because I didn't know
if I would be able to do it.
I started by drawing a circle
on the paper. Then I drew a
vertical line through the center
of the circle. Next, I drew a
horizontal line across the circle.
This created four equal quadrants.
I then drew diagonal lines from
the top-left and bottom-right
corners to the center of the circle.
This created eight equal sectors.
I continued this process until
I had drawn 16 equal sectors.
Finally, I filled in each sector
with a different color, creating
a colorful sunburst pattern.

$$S_1^3 = \{(1, 9), (9, 15), (11, 25), (14, 30)\}$$

$$S_1^4 = \{(0, 0), (8, 6), (10, 16), (13, 21), (1, 9), (9, 15)\}$$

$$S_1^4 = \{(0, 0), (8, 6), (10, 16), (13, 21), (1, 9), (9, 15), (11, 25), (14, 30)\}$$

$$\rightarrow S^4 = \{(0, 0), (8, 6), (10, 16), (13, 21), (1, 9), (9, 15)\}$$

$(13, 21)$ belongs to S_1^4 and also belongs to S_3

$$x^4 = 0 \quad x^3 = 1$$

Aggregation - merging left and right

left weight is equal to 1

middle is mid and the right is right

if $x^3 = 0$ then $x^4 = 0$

temp = $(0, 0, 0)$

$x^3 = 0$

$x^4 = 0$

$x^3 = 0$

$x^4 = 0$