UNIT-II

2. Greedy method:

1. Job sequencing with Deadlines:

consider a sequencing problem, where their are 4 jobs, those are J1, J2, J3 and J4

	J1	J2	J3	J4
Profit	50	15	10	25
Deadline	2	1	2	1

step1: Descending order of jobs based on Profit.

(J1, J4, J2, J3)

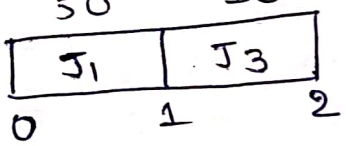
write corresponding deadlines

(2, 1, 1, 2)

step2: consider highest Deadline and find max Profit we get in that Deadline)x

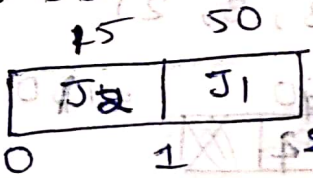
step2: Draw the gantt chart based on deadlines of the jobs.

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

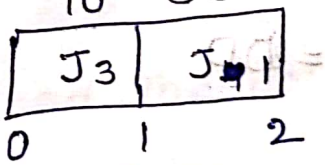


= 60

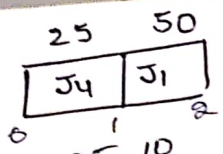
The combination of jobs J1 & J3 we can get the profit 60.



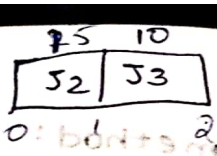
= 65



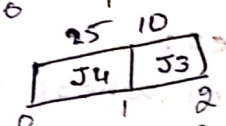
= 60



= 75



= 85



= 35

∴ we can get the maximum profit by selecting the sequence of jobs by ~~corres~~ considering deadlines.

max profit we can get = 75

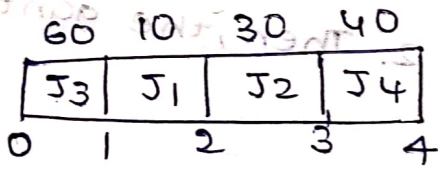
2.Q. consider a sequencing problems where their are 4 jobs having the profit of P(10, 30, 60, 40) and the corresponding deadlines d(2,3,1,4). find out max profit using greedy method where n=4

Sol: Descending order of jobs based on profits.

(J3, J4, J2, J1)

(1, 4, 3, 2)

Gantt chart:



= 140.

This is

J1 J2 J3 J4

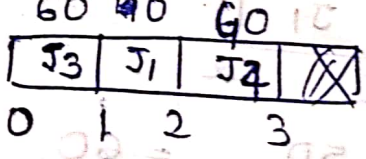
P(10,30,60,40)

d(2,3,1,3)

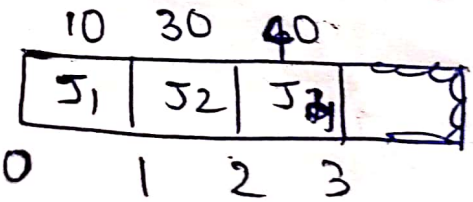
Descending order: (J3, J4, J2, J1)

(1, 3, 3, 2)

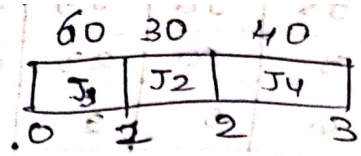
Gantt chart:



= 160



= 80



=130

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

max profit obtained = 130

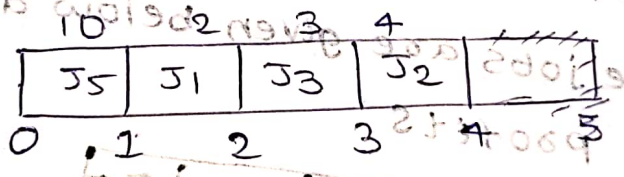
3Q. Find out the maximum profit by using the greedy method. the no. of jobs are 5. corresponding profits and weights are given below.

	J1	J2	J3	J4	J5
profit	2	4	3	1	10
weight	3	3	3	4	4

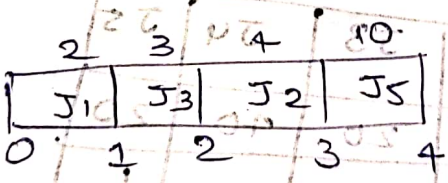
Sol: Descending order of jobs based on profits

(J5, J2, J3, J1, J4)
(10, 4, 3, 2, 1)

Gantt chart: (size = maximum deadline)



=19



=19

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

max profit obtained = 19.

4Q. Find out the maximum profit by using greedy method. The no. of jobs corresponding profits & weights are given below.

	J1	J2	J3	J4	J5	J6	J7	J8
Profit	15	20	30	18	18	10	23	16
weight	7	2	5	3	6	5	2	7

Sol: Descending order of Jobs based on profit

(J3, J9, J7, J2, J4, J5, J8, J1, J6)
(30, 25, 23, 20, 18, 18, 16, 15, 10)

Gantt chart:

20	23	25	18	30	15	16	=147
J2	J7	J9	J5	J3	J1	J8	

0 1 2 3 4 5 6 7

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

∴ we can get maximum profit by selecting the sequence of jobs (J2, J7, J9, J5, J3, J1, J8) by corresponding deadlines.

Maximum profit obtained = 147

5Q. 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
Dead line	2	1	3	2	1

Sol: Descending order of Jobs based on profit

(J2, J1, J4, J3, J5)

Gantt chart:

J2	J1	J3

0 1 2 3

maximum profit = 100+60+20
=180

∴ we can get the sequence corresponding maxim

6Q. Find out greedy meth along with

Jobs
Dead line
profit

Sol: Descend profit:

J

Gantt ch

maximur

∴ we can the sequ corresp maxi

Knap Sac

1Q. Find the problem, u object item, weights & p

P

W

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

maximum profit obtained = 180

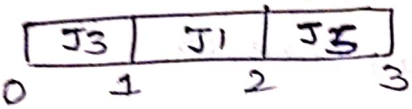
16. Find out maximum profit by using greedy method. All the jobs are given below along with deadline and profits.

Jobs	J1	J2	J3	J4	J5
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:



maximum profit = 27 + 100 + 15

= 142

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

maximum profit obtained = 142

Knap Sack Problem ← fractional KSP (greedy method) / 01KSP (Dynamic programming)

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

m - knapsack capacity

n - no. of items

	item 1	item 2	item 3
P	25	24	15
W	18	15	10

Sol: Greedy about weight:

Knapsnak weight m = 20

So, greedy about weight.

1st we can store/place item3. The item size is 10.

Now, the remaining bag size is 20-10=10.

Next, we can place item2. The item size is 15 but, the capacity of knapsack is only 10.

Maximum profit = 31

w	p
10	15
10	16
	31

Greedy about profit:

1st we can place item3. It's size is 18.

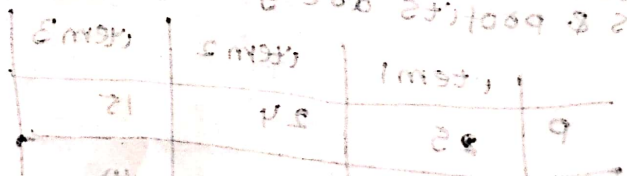
Now, remaining bag size is 20-18=2

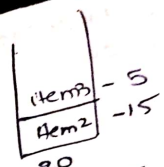
Next, we can place item2. It's size is 24 but the capacity of Knapsack is only 2.

w	p
18	25
2	24 $(\frac{2}{18})$
	3.2
	28.2

maximum profit is 28.2

Profit / weight :	item1	item2	item3
p	25	24	15
w	18	15	10
p/w →	$\frac{25}{18}$	$\frac{24}{15}$	$\frac{15}{10}$
	1.38	1.6	1.5
⇒	1.4	1.6	1.5





20-15=5

maxim

29. Find o

fractional

m=15, n

item is

obj	
p	
w	

step1: N

P/w rati

1st w

14. No

Next

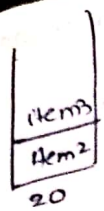
Si

is 9

It's s

size

i's



w	p
15	24
5	7.5
	31.5

1st we store item2. The item size is 15. Now, remaining bagsize is 20-15 = 5.

Next, we can place item3. The item size is 10 but, the capacity of knapsack is only 5.

maximum profit = 31.5

2Q. 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

step1: Now, calculate the p/w ratio

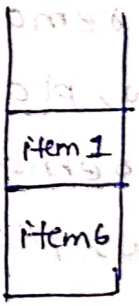
p/w ratio:

obj	1	2	3	4	5	6	7
p	10	5	15	7	6	18	3
w	2	3	5	7	1	4	1
p/w	5	1.66	3	1	6	4.5	3

Des Order of items based on p/w:

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

1st we store item 5. The item size is 1. Now, remaining bagsize 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 item6. It's size is 4. Now, remaining bag size is 8. place item 3. It's size is 5. Now, place item 7.



knapsack m=15

w	p	
4	18	15-4 = 14-12
2	10	14-2=9
4	18	12-4=8
3	15	8-5=3
1	6	3-1=2
2	3.33	

Max Profit = 55.33

38. Find a maximum profit by using greedy knapsack problem where $M=15$ and $n=5$ and each and every object profit and weight is given below.

obj	1	2	3	4	5
P	2	28	25	18	9
W	1	4	5	3	3

Sol: Given data,
The capacity of knapsack, $m=15$
The no. of units, $n=5$

Step1: find P/W ratio

obj	1	2	3	4	5
P	2	28	25	18	9
W	1	4	5	3	3
P/W	2	7	5	6	3

Step2: Descending order of objects based on P/W ratio.

2, 4, 3, 5, 2

Step3:
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
Now, place obj 5. It's size is 3

obj	W	P
2	15-4 = 11	28
4	11-3 = 8	18
3	8-5 = 3	25
5	3-3 = 0	9
		80

obj 5	3
obj 3	5
obj 4	3
obj 2	4
	15

max profit = 80

40. Find a max
problem wher
& wt is give

iten
w
p

Sol: Given d
Capacity
no.

Step1: P/W

obj
P
W
P/W

Step2 : Desc

Step3:
1st place obj
Now, rema
place, obj
Remaining

place o
Remainin
Now, pl

Maximu
30+

39. Find a maximum profit by using greedy knapsack problem where $M=15$ and $n=5$ and each and every object profit and weight is given below.

obj	1	2	3	4	5
P	2	28	25	18	9
W	1	4	5	3	3

Sol: Given data,

The capacity of knapsack, $m=15$

The no. of units, $n=5$

Step1: find p/w ratio

obj	1	2	3	4	5
P	2	28	25	18	9
W	1	4	5	3	3
P/W	2	7	5	6	3

Step2: Descending order of objects based on p/w ratio

2, 4, 3, 5, 2

Step3:

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

Now, place obj 5. It's size is 3

max profit = 80

obj 5	3	
obj 3	5	
obj 4	3	
obj 2	4	
	15	

obj	W	P
2	15-4 = 11	28
4	11-3 = 8	18
3	8-5 = 3	25
5	3-3 = 0	3 9
		80

Q8. Find a max profit by using greedy knapsack problem where M=60, n=5 and object profit & wt is given below.

item	1	2	3	4	5
w	5	10	15	22	25
p	30	40	45	77	90

Sol: Given data,

Capacity of knapsack, m=60

no. of units, n=5

step1: p/w ratio

obj	1	2	3	4	5
p	30	40	45	77	90
w	5	10	15	22	25
p/w	6	4	3	3.5	3.6

step2: Descending order of items based on p/w ratio.

1, 2, 5, 4, 3

step3:

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,

place obj 5, it's wt is 25

Remaining wt is 20

Now, place obj 4.

item	w	p
1	60-5	30
	55	
2	45	40
5	45-25	90
	20	
4	20	70
		130

Maximum Profit =

30+40+90+70 = 130

item 4	20
item 5	25
item 2	10
item 1	5

60

5Q. 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

Step1: 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

Step2: Descending order of objects based on P/w ratio

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

Step3:

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

Remaining size is 14

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

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 3 Remaining size is 2.

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

Max Profit = 51

obj	w	p
5	15-1=14	8
1	14-1=13	5
2	10	10
3	5	15
6	2	9
7	0	4

6Q. Find the m
knapsack prob
each & every ob

item	
P	
w	

Sol. Given d
knapsa
no. of

step1: find P

item	A
P	28
w	9
P/w	

step2: descen

step3: 1st p
Remaining
Now, place
Remainin
Now, pla
but we

Maximu

30
12
4 2/9

GQ. Find the max profit 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

Sol: Given data,
knapsack size, m = 60
no. of items = 4

step1: find P/W ratio

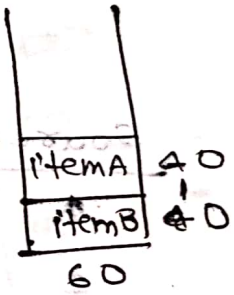
item	A	B	C	D
P	280	100	120	120
W	40	10	20	24
P/W	7	10	6	5

step2: descending order of items based on P/W ratio.

B, A, C, D

step3: 1st place item B. It's wt is 10.
Remaining wt 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.

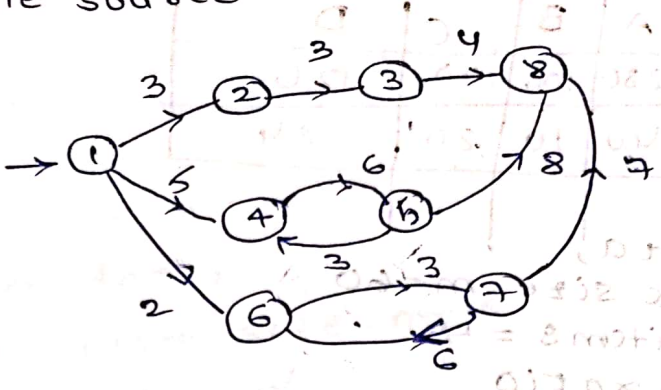
Maximum profit = 100 + 280 + 60
= 440.

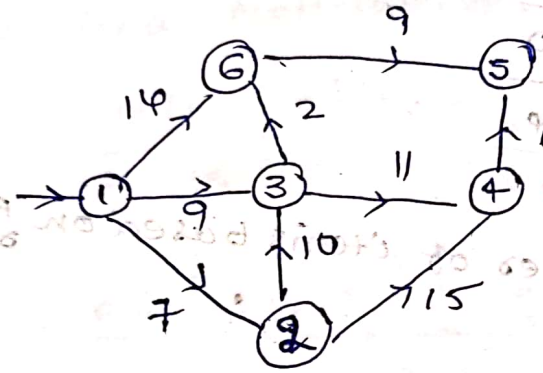


item	W	P
B	50	100
A	10	280
C	20	60
		440

Single Source Shortest path.

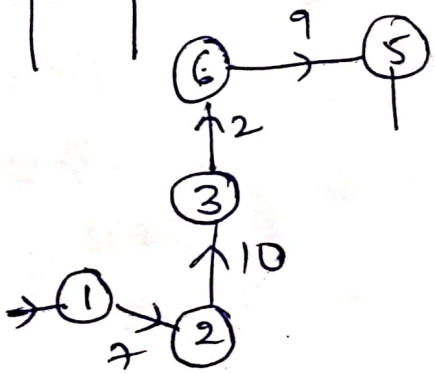
Eg: Find out the shortest path by using single source shortest path technique



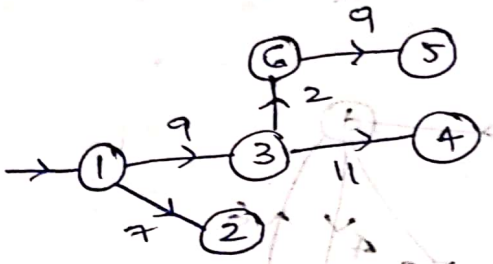


Source	2	3	4	5	6
1	7	9	∞	∞	16
1,2		17	22	∞	19
1,2,3			26	∞	
1,2,3,6			∞	28	
1,2,3,6,5			36		

1, 2, 3, 6, 5, 4.

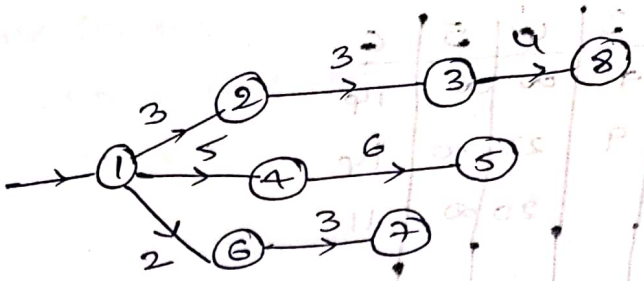


source	2	3	4	5	6
1	7	9	∞	∞	14
1, 2		9	22	∞	14
1, 2, 3			20	∞	11
1, 2, 3, 6			20	20	
1, 2, 3, 6, 4					
1, 2, 3, 6, 4, 5					

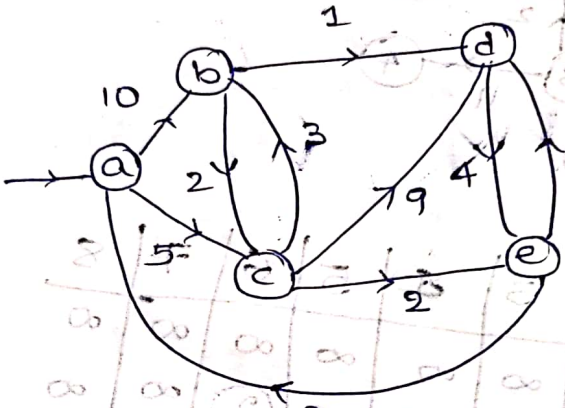


(Q1)

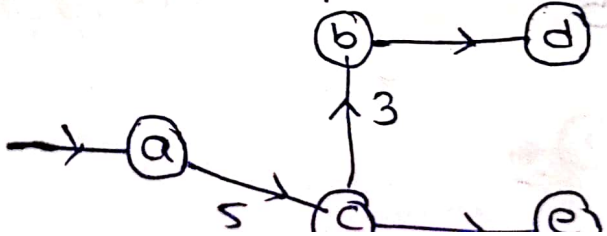
source	2	3	4	5	6	7	8
.	∞	∞	∞	∞	∞	∞	∞
1	3	∞	5	∞	2	∞	∞
1, 6	3	∞	5	∞		5	∞
1, 6, 2		6	5	∞		5	∞
1, 6, 2, 4		6		11		5	∞
1, 6, 2, 4, 7		6		11			12
1, 6, 2, 4, 7, 3				11			10
1, 6, 2, 4, 7, 3, 8				11			
1, 6, 2, 4, 7, 3, 8, 5							



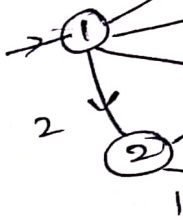
Q. Find out the shortest path by using single source shortest path technique.



source	b	c	d	e
	∞	∞	∞	∞
a	10	(5)	∞	∞
a,c	8	–	14	(7)
a,c,e	(8)	–	13	–
a,c,e,b			(9)	
a,c,e,b,d				



Q. Find out
source sh



source

1

1, 2

1, 2, 5

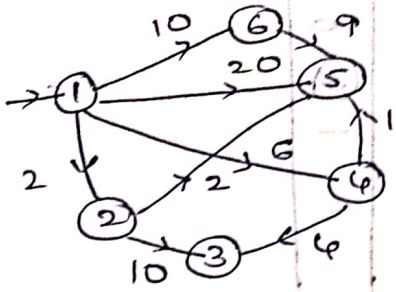
1, 2, 5, 4

1, 2, 5, 6

1, 2, 5, 6

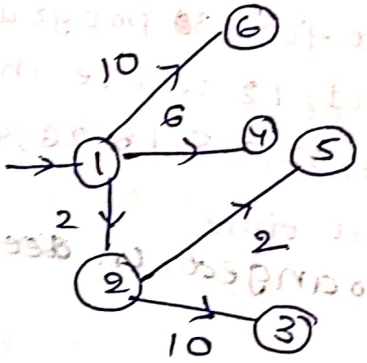
Q. Find
source

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

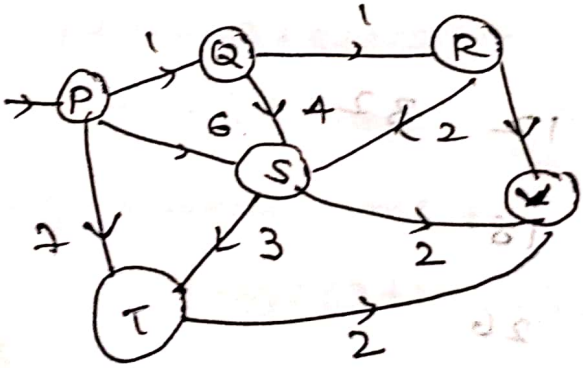


1, 2, 4, 3, 6, 5

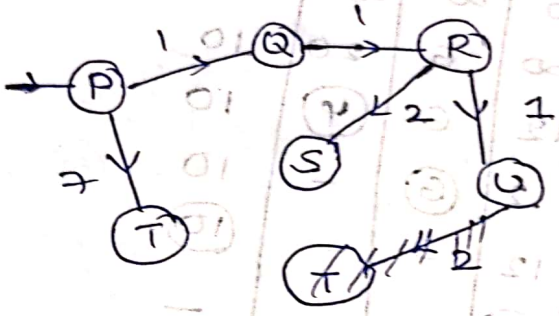
Source	2	3	4	5	6
	∞	∞	∞	∞	∞
1	(2)	∞	6	20	10
1, 2	—	12	6	(4)	10
1, 2, 5	—	12	(6)	—	10
1, 2, 5, 4 →	—	12	—	—	(10)
1, 2, 5, 6 →	—	(12)	—	—	—
1, 2, 5, 6, 3 →	—	—	—	—	—



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



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



Optimal Storage on Tapes using Greedy method.

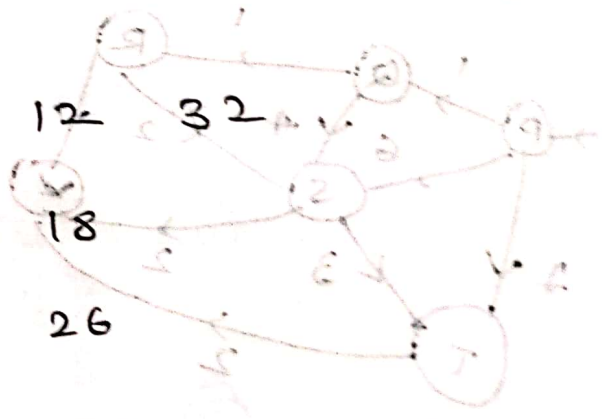
Q. Find the optimal placement for 13 programs on 3 tapes those are T0, T1, T2 where the program lengths are 12, 5, 8, 32, 7, 5, 18, 26, 4, 3, 10, 6. Find the optimal retreval time.

Step1: All the files are arranged in ~~decrea~~ ascending order.

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

Step2:

T0	3	5	8	12	32
T1	4	6	10	18	
T2	5	7	11	26	



step3: optimal retreval time for

$T_0 = 3 + (3+5) + (3+5+8) + (3+5+8+12) + (3+5+8+12+32)$

$= 3+8+16+28+60$

$= 115$

$T_1 = 4 + (4+6) + (4+6+10) + (4+6+10+18)$

$= 4+10+20+38$

$= 72$

$T_2 = 5 + (5+7) + (5+7+11) + (5+7+11+26)$

$= 5+12+23+49$

$= 89$

Average retreval time is

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

8. Find the optimal retreval time where n=10 and lengths of each files are 10, 20, 45, 70, 1, 3, 7, 54, 23, 67. There are 3 tapes.

step1: Ascending order of all files:

1, 3, 7, 10, 20, 23, 45, 54, 67, 70

step2: let us consider tapes as

T_0	1	10	45	70
T_1	3	20	54	
T_2	7	23	67	

step 3: finding optimal retreval time for T_0.

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

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

finding retreval time for T1.

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

$= 103$

finding retreval time for T2.

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

Step 4: Average retreval time =

$$\frac{194+103+134}{3} = \frac{431}{3} = 143.33 = 143$$

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

Sol: Step1: Arrange length of files in ascending order.

11, 12, 24, 34, 34, 34, 45, 56, 56, 73, 78, 91, 91

Step2: Let us consider tapes as.

T0	11	34	45	73	91
T1	12	34	56	78	
T2	24	34	56	91	

Step3: Finding retreval time for

T0 ⇒ 11 + (11+34) + (11+34+45) + (11+34+45+73) + (11+34+45+73+91)

⇒ 11 + 45 + 90 + 163 + 254

⇒ 563

T1 ⇒ 12 + (12+34) + (12+34+56) + (12+34+56+78)

⇒ 12 + 46 + 102 + 180

⇒ 340

T2 ⇒ 24 + (24+34) + (24+34+56) + (24+34+56+91)

⇒ 24 + 58 + 114 + 205

⇒ 401

Step4: Average retreval time =

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

Optimal

Q. Find out to merge for t

Ans Step1:

2, 3, 5,

merge 1

It is (n-

Ne 10,

It is no

Ascendin

1, 9

⇒ 1

Arrang

merg

Arran

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.

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

10, 7, 9, 13

It is not in Ascending order. Arange in Ascending order and merge 1st 2 files.

7, 9, 10, 13

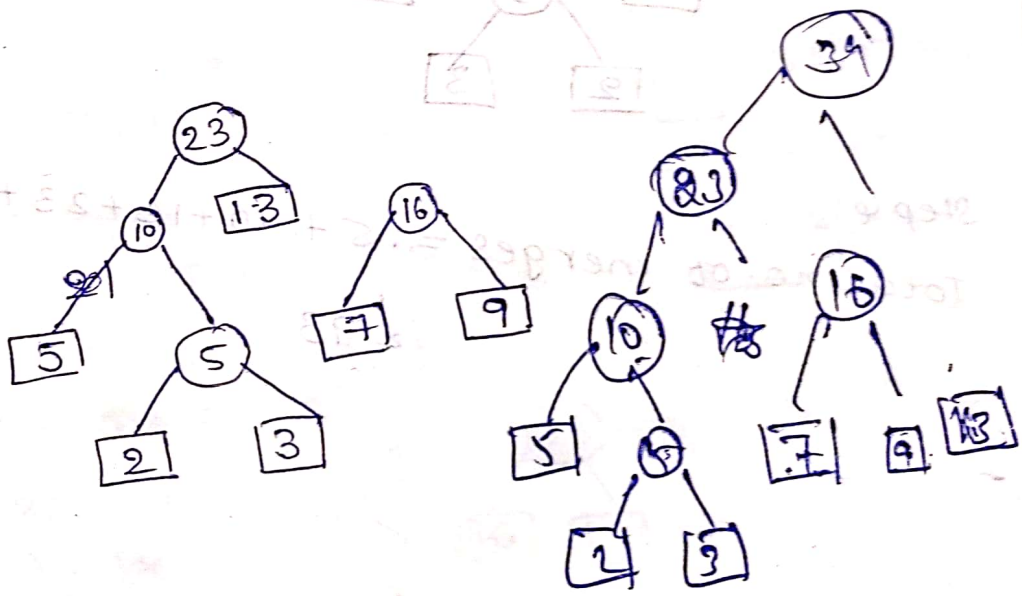
⇒ 16, 10, 13

Arrange in Ascending order: 10, 13, 16.

merge 1st 2 files: 23, 16.

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

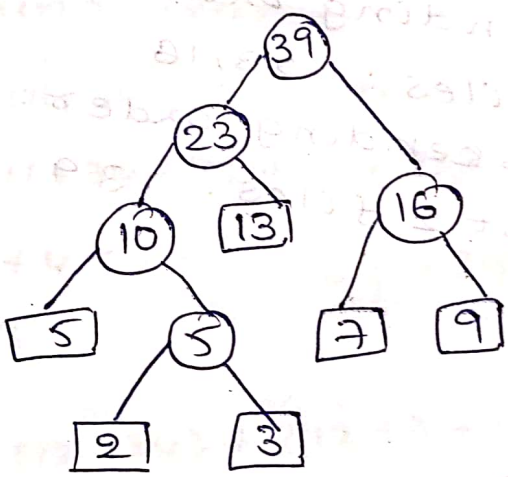
10, 13, 16

23, 16

16, 23

39

Step3: Construction of tree.



Step 4:

Total no. of merges = 5 + 10 + 16 + 23 + 39

= 93

Q. Find out to
set of files
Sol. Step 1: Ar
order. 2, 5
step2: Mer
list is in A
other wise
merge oper
left

2, 5, 7, 11

7, 7, 11

16, 11

11, 16

25

16

3

6. Find out total no. of merges for the set of files. 2, 5, 9, 26, 11, 16, 21, 17, 28, 42

Sol: Step 1: Arrange the given files in ascending order. 2, 5, 7, 11, 16, 17, 21, 26, 28, 42

Step 2: Merge 1st 2 files and if the obtained list is in ascending order, merge 1st 2 files otherwise Arrange in ascending order & perform merge operations repeat until only 1 element is left

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

7, 7, 11, 16, 17, 21, 26, 28, 42

16, 11, 16, 17, 21, 26, 28, 42

11, 16, 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

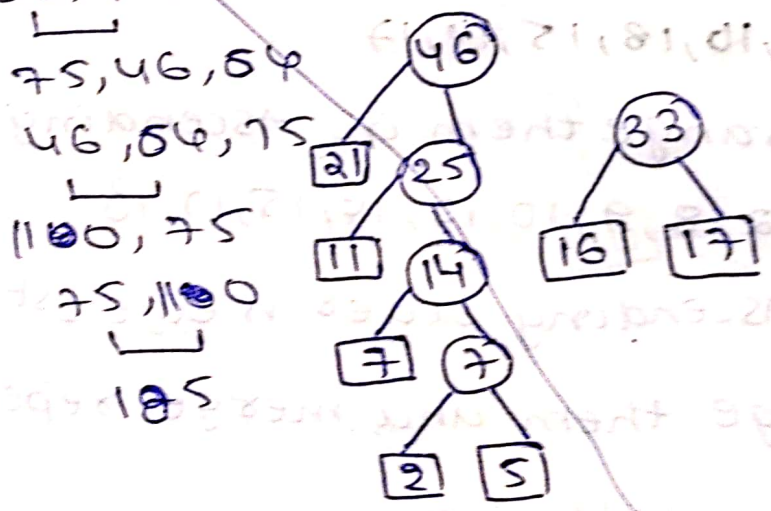
46, 54, 75

100, 75

75, 100

175

Step 3: Represent in the form of tree



26, 33, 38, 42, 46

59, 38, 42, 46

38, 42, 46, 59

80, 46, 59

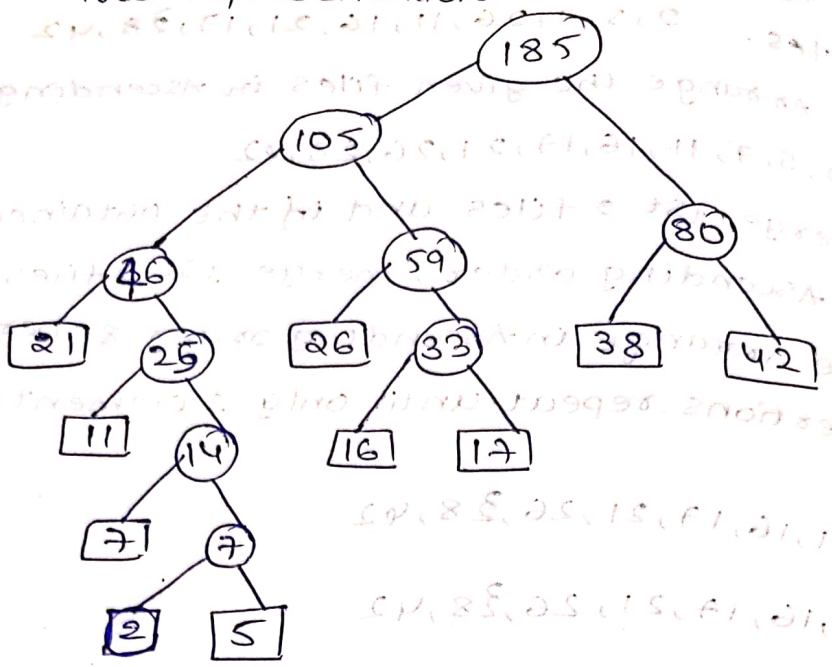
46, 59, 80

105, 80

80, 105

185

Tree Representation.



$\Sigma d_i x_i$

Step4: Total no. of merges =

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

$= 554$

we verify no. of merges using $\Sigma d_i x_i$

$6x2+6x5+5x7+4x11+21x3+16x4+17x4+$

$21x3+3x26+2x38+2x42$

$=554$

Q. Construct optimal merge tree for the given files

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

Sol: Step1: Arrange them in Ascending order.

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

Step2: If Ascending order, merge 1st two files. If not arrange them and merge. repeat until only one node will left

1, 2, 8, 9, 10, 12, 16, 15, 17, 18

3, 8, 9, 10, 12, 16, 15, 17, 18

11, 9, 10, 12, 16, 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

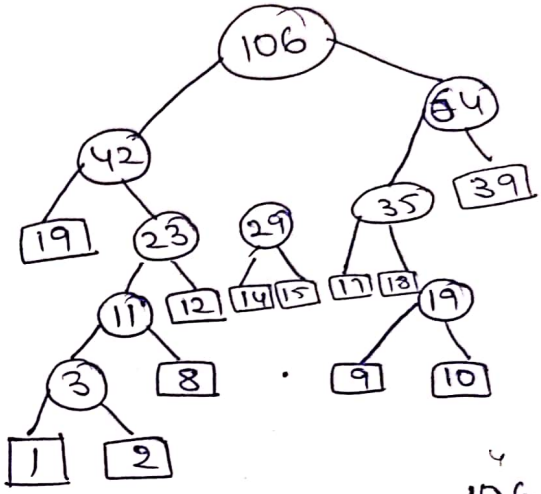
42, 29, 35

35, 29, 42

64, 42

42, 64

106



4
106
64
42
23
29
35
11
19
2 3
3 3 2

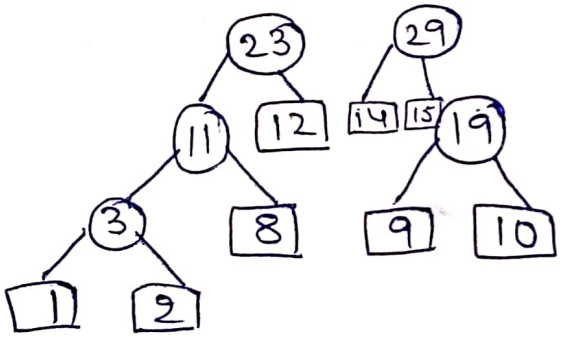
Step3: Tree Representation.

Step4:

Total no. of merges =

3 + 11 + 19 + 23 + 29 + 35 + 64 + 42 + 106

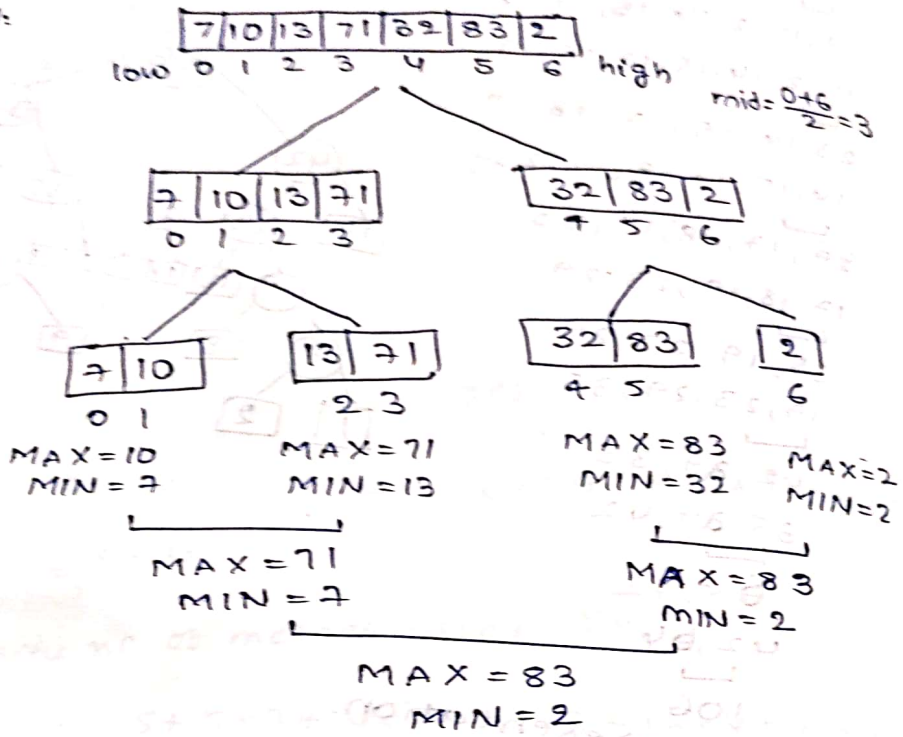
= 332



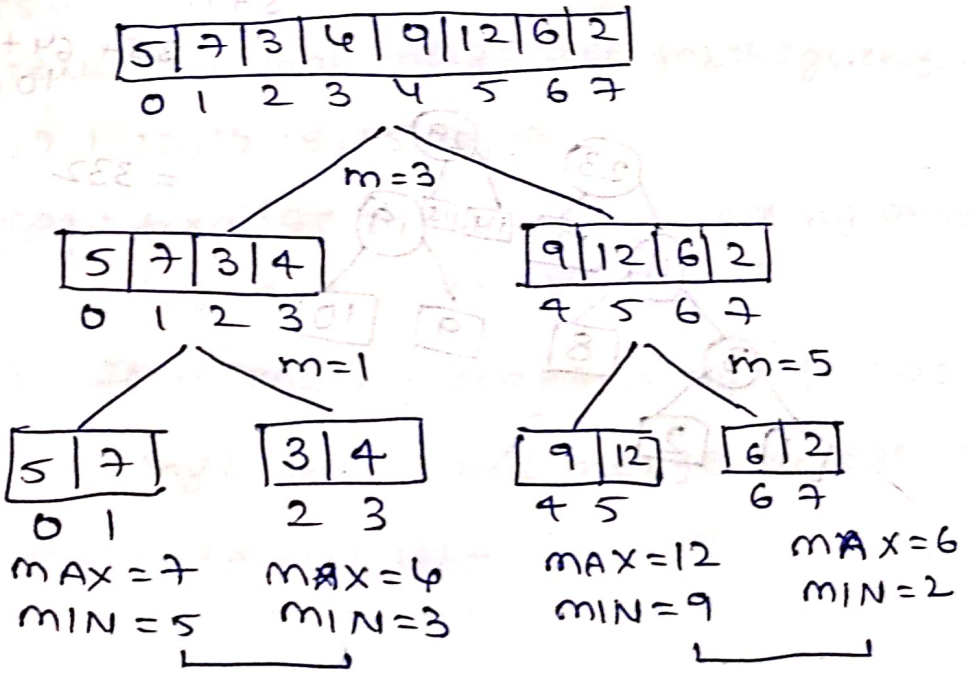
Finding a maximum & minimum element by using divide & conquer method.

Q. Find a maximum and minimum element in the given array. The array sizeis 7 (n=7).
The elements 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.



Algorithm:

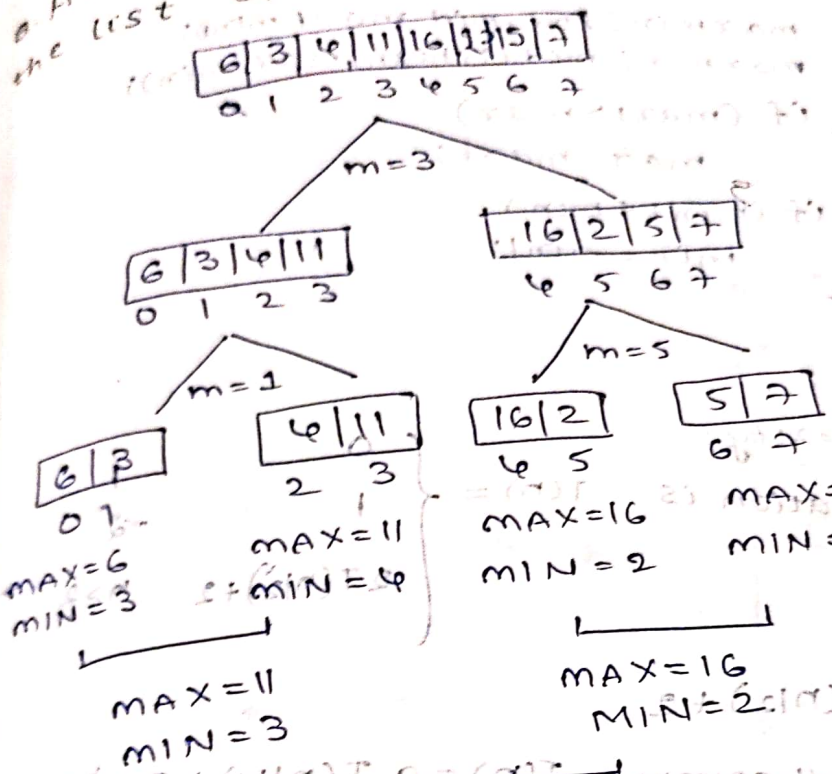
```
Algorithm
{
  if (low
  {
    max
  }
  else if
  {
    if
    {
```

max = 7
MIN = 3

MAX = 12
MIN = 2

MAX = 12
MIN = 2

e find a maximum & minimum element from the list 6, 3, 4, 11, 16, 2, 15, 7



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 = (low+high)/2;
  maxmin(low, mid, max, min);
  maxmin(mid+1, high, max1, min1);
  if (max1 > max)
      max = max1;
  if (min1 < min)
      min = min1;
}
}
```

Time Complexity:

Recurence relation is $T(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ 2T(n/2) + 2 & n > 2 \end{cases}$

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

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

$a = 2, b = 2, k = 0, p = 0$

$a > b^k$

$2 > 2^0$

$\log^p_n = 2$

$\Rightarrow n^2 = p$

use 1st formula.

$T(n) = \Theta(n^{\log^a_b})$

$= \Theta(n^{\log^2_2})$

$= \Theta(n)$

Using Back Substit

$T(n) = 2T(n$

$T(n) = 2(2$

$= 4$

$T(n) = 4$

$= 4$

$= 8$

$= 2^3$

$= 2^k$

$n/2^k$

2^k

$\Rightarrow$

$\Rightarrow$

$\Rightarrow$

Back-substitu

$T(n) = 2T($

$T(n-1) =$

$\Rightarrow$

$2^k T$

$\Rightarrow n/2$

$n =$

$\Rightarrow h$

Using Back Substitution method:

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

$T(n) = 2\,(2 \cdot T(n/4) + 2) + 2$

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

$T(n) = 4\,(2 \cdot T(n/8) + 2) + 4 + 2$

$= 8 \cdot T(n/8) + 8 + 4 + 2$

$\vdots$

$= 2^k \cdot T(n/2^k) + 2^k + 2^{k-1} + \cdots + 2^1$

$n/2^k = 1 \quad ; \quad n/2^k = 2$

$2^k = n \qquad 2^k = n/2$

$\Rightarrow n \cdot T(n/n) + 2^{k+1} \qquad \frac{n}{2} \cdot T(2n/n) + 2^k \cdot 2$

$\Rightarrow n(0) - 2^k \cdot 2 \qquad \frac{n}{2} - 2n$

$\Rightarrow -2n \qquad \frac{n - 4n}{2}$

$\Rightarrow O(n). \qquad \Rightarrow -3n/2$

Back-substitution method

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

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

$\Rightarrow 2^k \cdot T(n/2^k) + 2^k + 2^{k-1} + \cdots + 2$

$\Rightarrow 2^k\, T(n/2^k) + 4\left(1 - \frac{1}{2^k}\right)$

$\Rightarrow n/2^k = 1 \quad$ or $\quad n/2^k = 2.$

$n = 2^k \qquad n/2 = 2^k$

$\Rightarrow n\left(T(n/n) + 4\left(1 - \frac{1}{n}\right)\right) \Rightarrow n/2 \cdot T(2) + 4\left(1 - \frac{2}{n}\right)$

$\Rightarrow 0 + 4\left(\frac{n-1}{n}\right) \Rightarrow n/2 + 4n$

$2^k + 2^{k-1} + \cdots + 2$

$\Rightarrow \frac{2\,(1/2)^{k-1}}{1/2 - 1}$

$2 \cdot \frac{1}{2^{k-1}} \times \frac{2}{1}$

$2^2 \cdot 2^{k-1}$

$\Rightarrow 2^{k+1}$

$\frac{a(r^n - 1)}{r - 1}$

$r = 1/2, \; a = 2$

$n = k$

$\Rightarrow \frac{2(1/2^k - 1)}{1/2 - 1}$

$\Rightarrow \frac{2\left(1/2^k - 1\right)}{-1/2}$

$\Rightarrow 4(1 - 1/2^k)$

$n/2^k = 2$

$\Rightarrow 2^k = n/2$

$n/2 \cdot T(2^k/k) + 4\left(1 - \frac{2}{n}\right)$

$\Rightarrow \frac{n}{2} + 4 - \frac{8}{n}$

$\Rightarrow O(n)$

$\Rightarrow 2^k\left(T(n/2^k)\right) + \frac{a(r^n - 1)}{r-1}$ $\quad r=2,\ a=2,\ n=k$

$+ \frac{2.(2^k - 1)}{2-1}$

$\Rightarrow 2^{k+1} - 2$

$\Rightarrow 2^k\left(T(n/2^k)\right) + 2^{k+1} - 2$

$n/2^k = 2$

$2^k = n/2$

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

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

$\Rightarrow \frac{3n}{2} - 2$

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

Quick sort:

eg: 2, 9, 7, 5, 3, 1

pivot i

2	9	7
0	1	2

pivot i

2	9
0	1

pivot

2	9
0	1

pivot

2

2

2

Pivot=

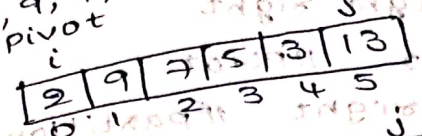
2

Pivo

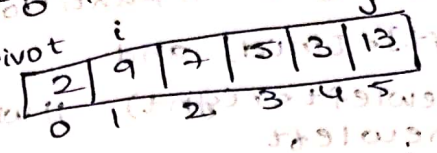
2

Quick Sort:

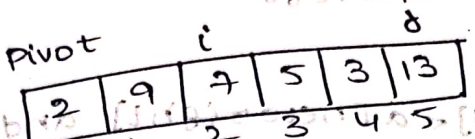
Eg: 2, 9, 7, 5, 3, 13.



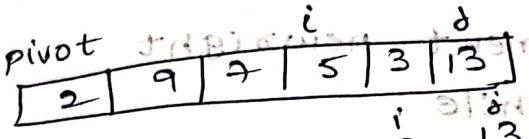
i=0 a[0] < a[j] 2 < 13 ✓ i++



a[1] < a[5] 9 < 13 ✓ i++



7 < 13 ✓ i++



5 < 13 ✓ i++

3 < 13 ✓ i++

i
3
j
13

2 9 7 5 3 13

i == j

j

Pivot = 5

2, 3

Pivot = 2

2, 3

9, 7, 13

Pivot = 7

2, 9, 7, 5, 3, 13

Pivot = 2

9, 7, 5, 3, 13

Pivot = 9

7, 5, 3

13

Pivot = 7

5, 3

Algorithm:

```
QuickSort(a, left, right):
  newleft = left, newright = right
  mid = a[left + right]/2   // Pivot
  while newleft ≤ newright   // partition
    while (a[newleft] < [mid] and
            newleft < right)   // while loop_1
      Increment newleft
    end while
    while (a[mid] < a[newright] and
            newright > left)   // while loop_2
      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:

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

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

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

$= 2[2T(n/$

$= 4T(n/4$

$= 4[2T($

$= 8T(n$

$= 8T($

$= 2^k \cdot T$

$n/2^k =$

$= nT$

$\Rightarrow n$

$T($

Time Complexi

Average Case

Worst Case

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

$T(n) = 2T(n/2) + n \quad ; n>1$

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

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

$= 4\left[2T(n/8) + \frac{n}{4}\right] + n + n$

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

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

$= 2^k \cdot T(n/2^k) + kn$

$n/2^k = 1 \Rightarrow 2^k = n$

$\log_2^n = k$

$= nT(n/n) + (\log_2^n)n$

$\Rightarrow n + n\log_2^n$

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

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(middle, right)
5. merge sort(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 nlogn (Avg, best, Worst)

UNIT-3

0-1 KNAPSACK Prob

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

formula:

Initially, $S^0 = (0,$

$S^{i+1} = S^i \cup S_1^i$

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

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

$S^3 = S^2 \cup$

$S^4 = S^3 \cup$

$S_1^0 = (2, 10)$

$S_1^0 = (0,0)$

$S^1 = (2, 10)$

$S^1 = \{(0,0)$ P, w

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

$S_1^1 = ((0,0)$

$= \{($

$S^2 = \{(0$

$S^2 = \{(0,$

$S^3 = S^2$

$S_1^2 = \{$

$= \{$

$S^3 =$

$S^3 =$

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

$S_1^3 = \{$

UNIT-3 2/2/23

0-1 KNAPSACK Problem:

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

formula: $S^{i+1} = S^i \cup S_1^i$

Initially, $S^0 = (0,0)$

$S^{i+1} = S^i \cup S_1^i$

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

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

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

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

$S_1^0 = (2,10)$

$S_1^0 = (0,0) + (2,10) = (2,10)$

~~$S^1 = (2,10)$~~

$S^1 = \{(0,0)\ (2,10)\}$ ($P_1\ w_1$, $P_2\ w_2$)

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

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

$= \{(5,15), (7,25)\}$ ($P_1 < P_2$, $w_1 > w_2$)

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

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

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

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

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

$S^3 = \{(0,0), (2,10), (5,15), (8,6), (10,16), (13,21)\}$ ($P_1\ w_1$, $P_2\ w_2$, $P_3\ w_3$, $P_4\ w_4$, $P_5\ w_5$)

(2,10) and (5,15) are X

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

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

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

$S_1^3 = \{ (1,9), (9,15), (11,25), (16,30) \}$

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

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

(13,21) belongs to S^4 and also belongs to S^3

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