B, B2
IT

24/05/21

Devansh Shah
1914078

Dshah

①

AA ESE

1) a
2) d
3) c
4) d
5) c
6) d
7) d

Devansh
1914078
Dshah ②

Q1) (b)

(i) Backtracking is a technique based on algorithm to solve a problem where it uses recursive calls to find the soln by building a soln step by step increasing values with time

Advantages:-

1) In comparison to D.P., this approach is more effective in some cases and the best option solving tactical problem.

2) Different states are stored into stack so data can be used anytime.

Dis adv:-

- runtime is slow
- need large amount of space
- detects conflicts too late

FOR EDUCATIONAL USE

Sundaram

Devranol
Devansh
194078
Dslah (3)

Q1) (b)

2) The elements are :-

(i) Substructure : decompose the given problem into smaller subproblems. Express the soln of the original problem in terms of solns for smaller problems. Unlike divide and conquer, there can be several decompositions which are different.

(ii) Table-Structure : After solving subproblems, store the answers to the subproblems in a table. This is done because subproblem solns are reused many times and we do not want to repeatedly solve the same problem over and over again.

(iii) Bottom-Up Computation : Using table to combine solns of smaller subproblems to solve larger subproblems and eventually arrive at a soln to complete problem. The idea goes like, bottom up means start with the smallest subproblems. Combining their solns obtain the solns to subproblems of increasing size until we arrive at the soln of the original problem.

Devansh
1914078

4

3) (i) In fractional knapsack problem, we are given weights and values of n items and we need to put these items in knapsack of capacity w to get the maximum total value in knapsack.

(ii) It is a maximization optimization problem which can be solved by greedy approach.

iii) The major difference between 0/1 knapsack and fractional knapsack problem is that in fractional knapsack, we can break items into pieces (fractions) where as 0/1 knapsack we ~~break~~ take one whole of the item or don't.

iv) The time complexity considering we sort the input first will be $O(n \log n)$ and the space complexity is $O(n)$ wher n : no. of items.

eg)

value	:	60	100	120
weight	:	10	20	30
fraction	:	6	5	4

capacity $w =$ ~~50~~ 50.

∴ Maximum value will be 240 by taking weight 10 and 20 kg and 2/3 fraction of 30 kg.

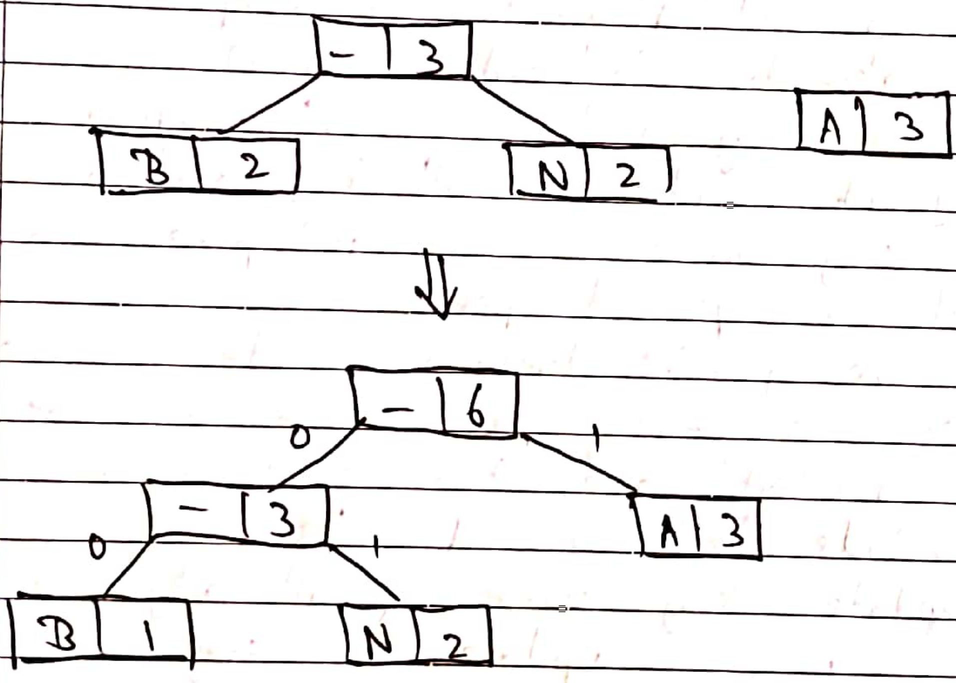
$$\therefore\ 60 + 100 + \frac{2}{3} \times 120 = 240$$

Devansh
1914078

Q1) b)

→ The unique characters in the string BANANA are 'B', 'N', 'A'.
Their frequencies are :-
B - 1
A - 3
N - 2

Generating Huffmann tree :-



Hence, B : 00, N : 01, A - 1

Devansh
1914078
Dslweh (6)

Q1) (b)

6) Prim's algorithm is a greedy algorithm that find a minimum spanning tree for a weighted undirected path.
It finds subset of the edges that form a tree that includes every index where the total weight of all edges in a tree is minimized.

Algorithm:
(1) Create a set 'finalSet' that keeps track of vertices already included in MST
(2) Assign a key value to all vertices in the input graph. Initialize all key values as infinite. Assign key value as 0 for first vertex so that it is picked first
(3) while the 'finalSet' doesn't include all vertices:
   a) Pick a vertex 'a' which is not there in 'finalSet' and has a minimum value key.
   b) Include 'a' to 'finalSet'
   c) Update key value of all adjacent vertices to 'a'. To update the key values, iterate through all if weight of edge 'a'-'b' is less than the previous key value of 'b'. Update the key value as weight of 'a'-'b'.

Time complexity: $O(E \log V)$
Space complexity: $O(E+V)$
E : edges (no. of)
V : vertices (no. of)

Devansh
1914078
Dshah 7

Q 2) 1)

i) $T(n) = 27\,T(n/3) + n$

$T(n) = aT(n/b) + \theta(n^k \log^p n)$ ⟶ Master's theorem

$a = 27 \quad b = 3 \quad k = 1 \quad p = 0$

$b^k = 3^1 = 3$

$\because a > b^k$

$\therefore T(n) = \theta(n^{\log_b a})$

$= \theta(n^{\log_3 27})$

$= \theta(n^{\log_3 3^3}) \quad \because \log_b a^c = c \log_b a$

$= \theta(n^3)$

Ans $T(n) = \theta(n^3)$

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

$T(n) = aT(n/b) + \theta(n^k \log^p n)$ ⟶ Master's theorem

$a = 2$

$b = 2$

$k = 1 \quad p = 0$

$b^k = 2^1 = 2$

$\because a = b^k \;\&\; p > -1$

$T(n) = \theta(n^{\log_b a} \log^{p+1} n)$

$= \theta(n^{\log_2 2} \log n)$

$= \theta(n \log n)$

Ans) $T(n) = \theta(n \log n)$

Devansh
1914078

8

Q2) 2) Solving the following list using radix sort
23 34 67 89 123 63 39 212 90

In radix sort, the sort is performed based on the place value of number and number of steps in the process is determined by the maximum no. of digits in the given list
Here, 212 = 3 digits i.e. 3 steps

To sort, the numbers are divided into buckets from 0 - 9 in every step. Add 0 to make 3 digit

Step 1 :- Unit's place
- we place numbers in resp buckets according to the digit in unit's place

0 : 090
1 :
2 : 212
3 : 023, 123, 063
4 : 034
5 :
6 :
7 : 067
8 :
9 : 089, 039

Take out numbers from top to bottom in order, we get
090, 212, 023, 123, 063, 034, 067, 089, 039

Devansh
1914078
Dalruh (9)

Step 2 :- Applying radix sort on step 1's output
- Ten's place

0 :
1 : 212
2 : 023 , 123
3 : 034 , 039
4 :
5 :
6 : 063 , 067
7 :
8 : 089
9 : 090

We get,

212 , 023, 123, 034, 039, 063 , 067, 089, 090

Step 3 : - Hundredth's place

0 : 023 , 034 , 039 , 063 , 067 , 089 , 090
1 : 123
2 : 212
3 :
4 :
5 :
6 :
7 :
8 :
9 :

Devansh
1914078
Delsh 10

Final answer after sorting:

023, 034, 039, 063, 067, 089, 090, 123, 212

$\Rightarrow$ i.e. 23, 34, 39, 63, 67, 89, 90, 123, 212

Devansh
1914078
Devansh

Q3) Solving matrix chain multiplication by D.P. iterative (table) approach.

Formula used : $m[i,j] = \min\{ \min[i,k] + \min[k+1,j] + d_{i-1} * d_k * d_j \}$

where $*$ is $i \leq k < j$

$m[i,j]$ values at ith row, jth column and d_i dimension.

given : $\langle \overset{d_0}{4}, \overset{d_1}{10}, \overset{d_2}{3}, \overset{d_3}{12}, \overset{d_4}{20}, \overset{d_5}{7} \rangle$

$\therefore$ $\underset{4\times10}{A_1}$ $\underset{10\times3}{A_2}$ $\underset{3\times12}{A_3}$ $\underset{12\times20}{A_4}$ $\underset{20\times7}{A_5}$

<u>Step 1</u> : Make a diagonal matrix and fill the main diagonal as 0

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & \\ \hline 0 & & & & & 1 \\ & 0 & & & & 2 \\ & & 0 & & & 3 \\ & & & 0 & & 4 \\ & & & & 0 & 5 \end{array}$$

<u>Step 2</u> : product cost of 2 matrices by taking $j-1=1$

$\therefore m[1,2] = d_0 d_1 d_2$

$= 4 \times 10 \times 3$

$= 120$

Devansh
1914078

12

$\therefore m[2,3] = d_1 d_2 d_3$
$= 10 \times 3 \times 12$
$= 360$

$m[3,4] = d_2 d_3 d_4$
$= 3 \times 12 \times 20$
$= 720$

$m[4,5] = d_3 d_4 d_5$
$= 12 \times 20 \times 7$
$= 1080$

Filling table with these values

	1	2	3	4	5
1	0	120			
2		0	360		
3			0	720	
4				0	1080
5					0

Making additional table to note key matrix whose multiplication is taking place so we can trace back

	1	2	3	4	5
1		1			
2			2		
3				3	
4					4
5					

Devansh
1914078
13

Step 3: same steps for 3 matrices together.

$m[1,3] = \min[m[1,2] + m[2,3] + d_0 d_1 d_3,\ m[1,2], m[3,3] + d_0 d_2 d_3]$
$= \min[840, 264]$
$= 264$ at matrix 2

$m[2,4] = \min[2760, 1320]$
$= 1320$ at matrix 2

$\min[2,5] = \min[1140, 1932]$
$= 1140$ at 4

C.

1	2	3	4	5	
0	120	264			1
	0	360	1320		2
		0	~~1320~~ 720	1140	3
			0	1680	4
				0	5

S4) Repeating for 4 matrices together

$$m[1,4] = \min\begin{cases} m[1,1] + m[2,4] + d_0 d_1 d_4 = 2120 \\ m[1,2] + m[3,4] + d_0 d_2 d_4 = 1080 \\ m[1,3] + m[4,4] + d_0 d_3 d_4 = 1224 \end{cases}$$

$= 1080$

FOR EDUCATIONAL USE
Sundaram

Devansh
1914078
14 Dshah

$m[2,5] = \min[1350, 2380, 1720]$
$= 1350$

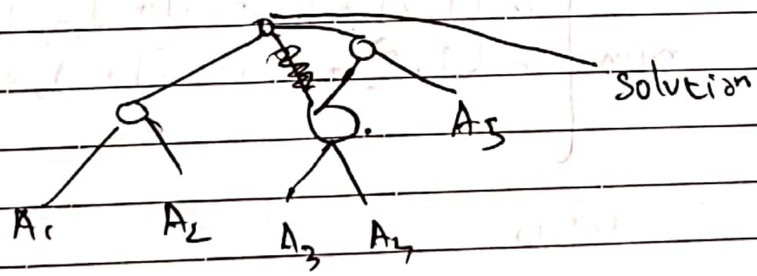
	1	2	3	4	5	
	0	120	264	1080		1
		0	360	1320	1350	2
			0	720	1140	3
				0	1080	4
					0	5

$m[1,5] = 1344$

$\therefore$

	1	2	3	4	5	
	0	120	264	1080	1344	1
		0	360	1320	1350	2
			0 ~~720~~	720	1140	3
				0	1080	4
					0	5

The min cost of multiplication of 5 matrices is 1344.
and in order
$(A_1, A_2)(A_3, A_4, A_5)$

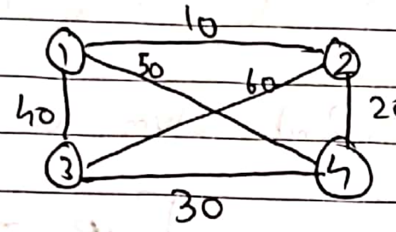


derived from 2nd table

FOR EDUCATIONAL USE
Sundaram

Devansh
1914078
15

Q4) Travelling salesman problem using branch & bound



The problem states that we will make a state space tree in such a way that salesman will start and end at the same node at the minimum cost and will travel through all n nodes in the process.

Make initial matrix, start from 1
∴ minimization problem
∴ branch and bound

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} \infty & 10 & 40 & 50 \\ 10 & \infty & 60 & 20 \\ 40 & 60 & \infty & 30 \\ 50 & 20 & 30 & \infty \end{bmatrix} \end{array}$$

Reducing the matrix by subtracting every row by min of that row and every column by min of column

row   col

$$\begin{array}{c} \begin{bmatrix} \infty & 0 & 30 & 40 \\ 0 & \infty & 50 & 10 \\ 10 & 30 & \infty & 0 \\ 30 & 0 & 10 & \infty \end{bmatrix} \begin{array}{c} 10 \\ 10 \\ 30 \\ 20 \end{array} \\ \begin{matrix} 0 & 0 & 10 & 0 \end{matrix} \quad 70+10 \\ =80 \end{array} = \begin{bmatrix} \infty & 0 & 20 & 40 \\ 0 & \infty & 40 & 10 \\ 10 & 30 & \infty & 0 \\ 30 & 0 & 0 & \infty \end{bmatrix}$$

reduced cost = 8   upper = ∞

Devansh
1914078
16 Dshah

We now draw matrix for sales man if goes from 1 to 2, 3 4 & their reduced matrix cost

1 - 2 1st row & 2nd colum will be ∞

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 40 & 10 \\ 10 & \infty & \infty & 0 \\ 30 & \infty & 0 & \infty \end{bmatrix} \begin{matrix} \\ 10 \\ 0 \\ 0 \end{matrix} = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 30 & 0 \\ 0 & \infty & \infty & 0 \\ 20 & \infty & 0 & \infty \end{bmatrix}$$

10 $\hat{r} = 10 + 10 = 20$

cost of 2 will be $c(1,2) + r + \hat{r} = 0 + 80 + 20 = 100$

1 - 3

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 10 \\ \infty & 30 & \infty & 0 \\ 3 & 0 & \infty & \infty \end{bmatrix} \begin{matrix} \\ 0 \\ 0 \\ 0 \end{matrix}$$

0 0 0 $\hat{r} = 0$

cost of 3 will be $c(1,3) + r + \hat{r} = 20 + 80 + 0 = 100$

1 - 4

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 40 & \infty \\ 10 & 30 & \infty & \infty \\ \infty & 0 & 0 & \infty \end{bmatrix} \begin{matrix} \\ \\ 10 \\ \end{matrix} = \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 40 & \infty \\ 0 & 20 & \infty & \infty \\ \infty & 0 & 0 & \infty \end{bmatrix}$$

$\hat{r} = 10$

cost of 4 will be $c(1,4) + r + \hat{r} = 30 + 80 + 10 = 120$

Sundaram
FOR EDUCATIONAL USE

Devansh
1914078

As in B&B we follow min cost, we continue with 2 & 3 as same cost

We can only go 2-3, 2-4 $r = 100$

2 - 3

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 20 & \infty & \infty & \infty \end{bmatrix} 20 = \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & \infty \end{bmatrix}$$

$\hat{r} = 20$

cost of 3 → $c(2,3) + r + \hat{r} = 30 + 20 + 100 = \underline{150}$

2 - 4

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty \end{bmatrix} \begin{matrix} \\ \\ 0 \\ 0 \end{matrix}$$

$\hat{r} = 0$

cost of 4 → $c(2,4) + r + \hat{r} = 0 + 0 + 100 = 100$

// we do for 3 if we choose 3 instead of 2

$r = 100$

3 - 2

cost of 2 = 170

3 - 4

cost of 4 = 130

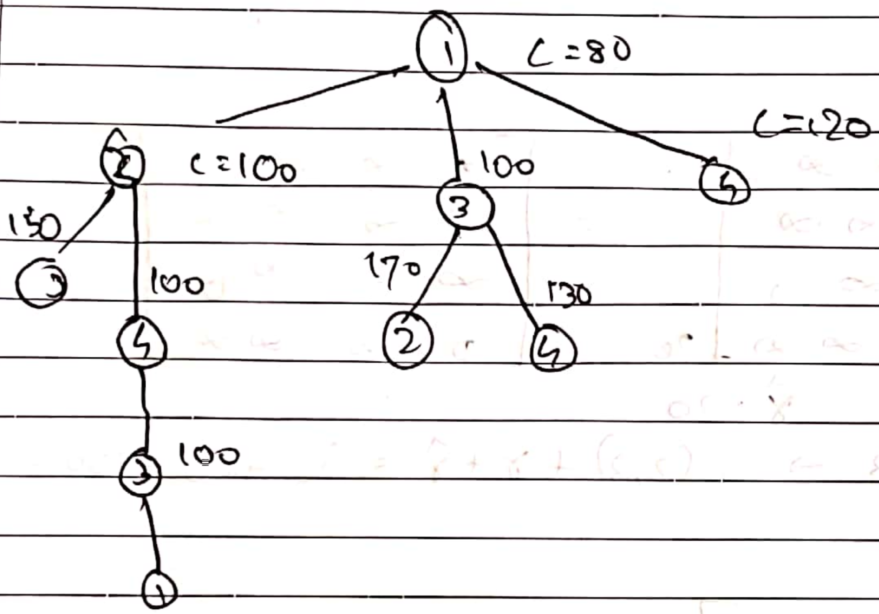
As min cost among all is 2 - 4 we will continue with that.

Sundaram
FOR EDUCATIONAL USE

Devansh
1914078

18

4 - 3
cost of 3 = 100
$\therefore$ 3 is minimum cost, so we will go with that
we will finally connect 3 - 1 to complete cycle.



space state tree

To get upper,
1 - 2 - 4 - 3 - 1
upper = 10 + 20 + 30 + 40 = 100

so minimum cost = 100 by following the path 1, 2, 4, 3, 1