Student Name : Bazgha Razi

College Name : DGIT

Roll Number : 191380214

Subject Name : DAA

Subject Code : PCC-CSE-307G

Course : B. Tech

Semester : 5

No. of pages : 12

Student Signature : Bazgha Razi

Date of Examination : 16/12/2021

Name: Bazgha Page No: 1 Date: 16/12/21 Signature: Bazgha Razi

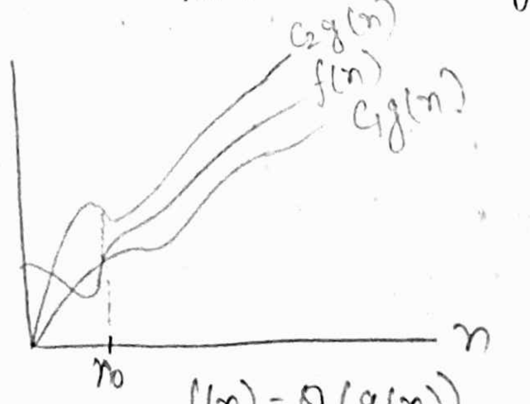
Ans 1a) Asymptotic Notations

It is used to describe the time to run an algorithm. There are various notation such as Θ, O, Ω, etc.

i) Θ - notation

For function $g(n)$, we define $\Theta(g(n))$ big-theta of n, as the set

$\Theta(g(n)) = \{f(n) : \exists$ positive constants c_1, c_2 and n_0, such that $\forall\ n \geq n_0$. We have $0 < c_1 g(n) \leq f(n) \leq c_2 g(n)\}$

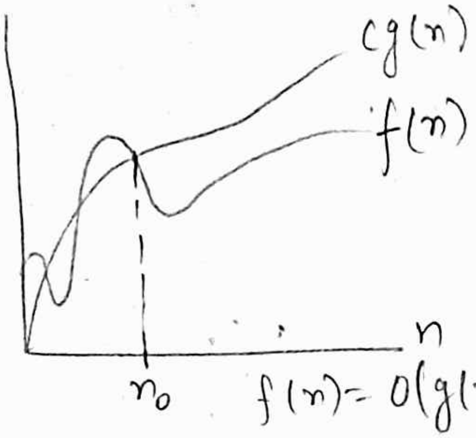


$f(n) = \Theta(g(n))$

ii) O - notation

For function $g(n)$, we define $O(g(n))$ big O of n, the set:

$O(g(n)) = \{f(n) : \exists$ positive constants c and n_0, such that $\forall\, n \geq n_0$ we have $0 \leq f(n) \leq c\,g(n)\}$

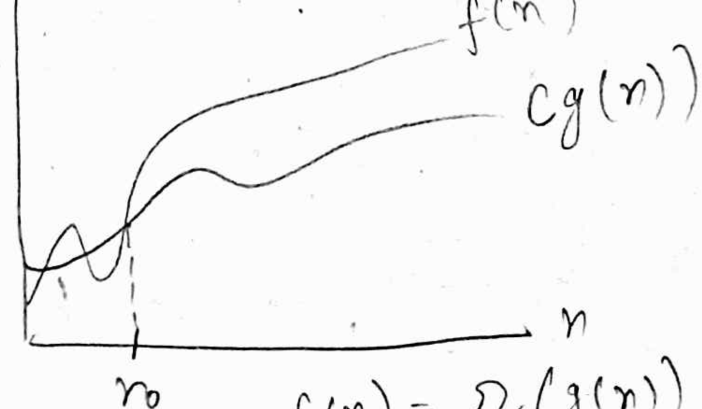


$f(n) = O(g(n))$

iii) Ω-notation

For function $g(n)$, we define $\Omega(g(n))$, big-omega of n, as the set:

$\Omega(g(n)) = \{f(n) : \exists$ positive constants c and n_0, such that $\forall\, n \geq n_0$.
We have $0 \leq c\,g(n) \leq f(n)\}$



$f(n) = \Omega(g(n))$

Name: Bazgha Page No: 3 Date: 16/12/21 Signature: Bazgha Razi

Ans 1b) Dynamic Programming

It is an algorithmic technique for solving by recursively breaking down into simpler subproblems and using fact that the optimal solution to the overall problem depends upon the optimal solution to it's individual problems or subproblems.

Dynamic programming approach was developed by Richard Bellmam in 1950s.

This algorithm solves each problem or subproblem just once and then remembers its answer, thereby avoiding re-computation of the answer for similar subproblem every time.

Application of dynamic programming

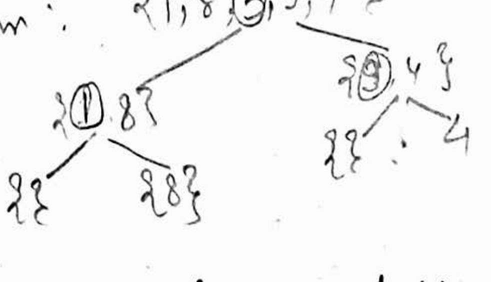
* It is used for 0/1 knapsack problem.
* It is used for optimal binary search tree.
* It helps in mathematical optimization problem.
* It helps to find shortest path.
* It also help in robotics control.

Ans 2a) Complexity analysis for quick sort, merge sort and selection sort are

Quick Sort: It is a sorting algorithm which uses divide and conquer technique. In this, we choose an element as pivot and then create partition of array arround the pivot element by repeating the technique for each subpart and sort accordingly.

Best case of quick sort is when we select pivot as a mean element. In this case, the recursion will look as shown in diagram the height of tree is logN and in each level we will be traversing to all the elements with total operation will be logN*N

Diagram:



Time complexity: O(NlogN)

Worst Case Time Complexity: $O(N^2)$

$T(n) = T(k) + T(n-k-1) + cn$

When the array is sorted or reverse sorted, the partition algorithm divides the array in 2 subarrays with 0 and n-1 elements.

Therefore, $T(n) = T(0) + T(n-1) + cn$

On solving we get,

$T(n) = O(n^2)$

On an average case, the partition algorithm divides the array into two subarrays with equal size. Therefore,

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

On solving we get,

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

Name: Bazgha Page No: 6 Date: 16/12/21 Signature: Bazgha Kazi

Merge Sort: It is also based on divide and conquer technique. First we have to dive the array into subarrays and then sort them and after sorting, we have to combine then again in the sorted form.

Time Complexity: O(NlogN)

Selection Sort: In this, algorithm will first find the smallest element in the array and swap it with the element in the first position. and Then it will check for second element and swap it with the next smallest element. This technique is repeated until the array is completed sorted.

~~Worst~~ Worst Case Complexity: $O(n^2)$

Best Case Complexity: $O(n^2)$

Ans 3a) Optimal Binary Search Tree

	0	1	2	3	4
node		10	20	30	40
p_i		3	3	1	1
q_i	2	3	1	1	1

p_i → for successful search probability
q_i → for unsuccessful search probability

$$c[i,j] = \min_{i<k\le j}\{c[i,k-1] + c[k,j]\} + w[i,j]$$

$$w[i,j] = w[i,j-1] + p_j + q_j$$

	0	1	2	3	4
0	$w_{00}=2$ $c_{00}=0$ $r_{00}=0$	$w_{11}=3$ $c_{11}=0$ $r_{11}=0$	$w_{22}=1$ $c_{22}=0$ $r_{22}=0$	$w_{33}=1$ $c_{33}=0$ $r_{33}=0$	$w_{44}=1$ $c_{44}=0$ $r_{44}=0$
1	$w_{01}=8$ $c_{01}=8$ $r_{01}=1$	$w_{12}=7$ $c_{12}=7$ $r_{12}=2$	$w_{23}=3$ $c_{23}=3$ $r_{23}=3$	$w_{34}=3$ $c_{34}=3$ $r_{34}=4$	
2	$w_{02}=12$ $c_{02}=19$ $r_{02}=1$	$w_{13}=9$ $c_{13}=12$ $r_{13}=2$	$w_{24}=5$ $c_{24}=8$ $r_{24}=3$		
3	$w_{03}=14$ $c_{03}=25$ $r_{03}=2$	$w_{14}=11$ $c_{14}=19$ $r_{44}=2$			
4	$w_{04}=16$ $c_{04}=32$ $r_{04}=2$				

Name: Bazgha Page No: 8 Date: 16/12/21 Signature: Bazgha Razi

$W_{00} = q_0 = 2$

$W_{11} = q_1 = 3$

$W_{22} = q_2 = 1$

$W_{33} = q_3 = 1$

$W_{44} = q_4 = 1$

using, $w[i, j-1] + p_j + q_j$, we have to find the remaing weight

$\therefore$ $w[0,1] = w[0,0] + p_1 + q_1$
$i=0, j=1$ $= 2 + 3+3 = 8$

$w[0,2] = w[0,1] + p_2 + q_2$
$= 8 + 3 + 1 = 12$

$w[1,2] = w[1,1] + p_2 + q_2$
$= 3 + 3 + 1 = 7$

$w[2,3] = w[2,2] + p_3 + q_3$
$= 1 + 1 + 1 = 3$

$w[3,4] = w[3,3] + p_4 + q_4$
$= 1 + 1 + 1 = 3$

$w[0,2] = w[0,1] + p_2 + q_2$
$= 8 + 3 + 1 = 12$

$w[1,3] = w[1,2] + p_3 + q_3$
$= 7 + 1 + 1 = 9$

$w[2,4] = w[2,3] + p_4 + q_4$
$= 3 + 1 + 1 = 5$

$w[0,3] = w[0,2] + p_3 + q_3$
$= 12 + 1 + 1 = 14$

$w[1,4] = w[1,3] + p_4 + q_4$
$= 9 + 1 + 1 = 11$

$w[0,4] = w[0,3] + p_4 + q_4$
$= 14 + 1 + 1 = 16$

Now find cost for the given OBST :

$c[0,0] = 0$
similarly; c_{11}, c_{22}, c_{33}, $c_{44} = 0$

$c[0,1] = \min_{0 < k \leq 1} \{ c[0,0] + c[1,1] \} + w[0,1]$
$k = 1$
$= \min\{0 + 0\} + 8$
$c[0,1] = 0 + 8 = 8$

Similarly,

$$C[1,2] = \min_{1<k\le 2} \{C[1,1] + C[2,2]\} + w[1,2]$$

$k=2$

$$= 0 + 7 = 7$$

$$C[2,3] = 3$$

$$C[3,4] = 3$$

$$C[0,2] = \min_{0<k\le 2} \{ \underbrace{(C[0,0] + C[1,2])}_{k=1}, \underbrace{(C[0,1] + C[2,2])}_{k=2} \} + w[0,2]$$

$k = 1, 2$

$$= \min\{(0+7), (8+0)\} + 12$$

$$= 7 + 12 = 19$$

$$C[1,3] = \min_{1<k\le 3} \{ \underbrace{(C[1,1] + C[2,3])}_{k=2}, \underbrace{(C[1,2] + C[3,3])}_{k=3} \} + w[1,3]$$

$k = 2, 3$

$$= \min\{(0+3), (7+0)\} + 9$$

$$= 3 + 9 = 12$$

$$C[2,4] = \min_{2<k\le 4} \{ \underbrace{(C[2,2] + C[3,4])}_{k=3}, \underbrace{(C[2,3] + C[4,4])}_{k=4} \} + w[2,4]$$

$k = 3, 4$

$$= \min\{(0+3), (3+0)\} + 5$$

$$= 3 + 5 = 8$$

$$C[0,3] = \min_{\substack{0<k\le 3 \\ k=1,2,3}} \left\{ \underbrace{(C[0,0]+C[1,3])}_{k=1}, \underbrace{(C[0,1]+C[2,3])}_{k=2}, \underbrace{(C[0,2]+C[3,3])}_{k=3} \right\} + w[0,3]$$

$$= \min\{(0+12), (8+3), (19+0)\} + 14$$

$$= 11 + 14 = 25$$

$$C[1,4] = \min_{\substack{1<k\le 4 \\ k=2,3,4}} \left\{ \underbrace{(C[1,1]+C[2,4])}_{k=2}, \underbrace{(C[1,2]+C[3,4])}_{k=3}, \underbrace{(C[1,3]+C[4,4])}_{k=4} \right\} + w[1,4]$$

$$= \min\{(0+8), (7+3), (12+0)\} + 11$$

$$= 8 + 11 = 19$$

$$C[0,4] = \min_{\substack{0<k\le 4 \\ k=1,2,3,4}} \left\{ \underbrace{C[0,0]+C[1,4]}_{k=1}, \underbrace{C[0,1]+C[2,4]}_{k=2}, \underbrace{C[0,2]+C[3,4]}_{k=3}, \underbrace{C[0,3]+C[4,4]}_{k=4} \right\} + w[0,4]$$

$$= \min\{(0+19), (8+8), (19+3), (25+0)\} + 16$$

$$= 16 + 16$$

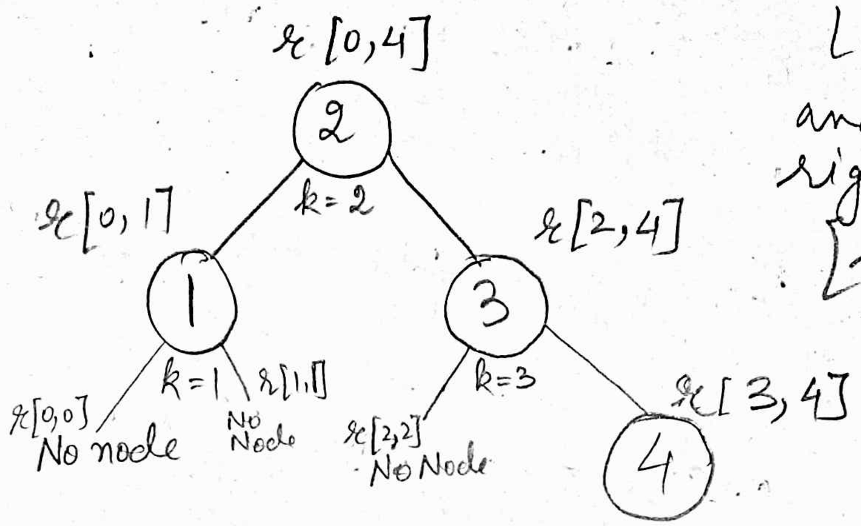
$$= 32$$

Now, we have to create optimal binary search tree,

So, $r[0,4] = 2$

Name : Bazgha   Page No : 12 Date : 16/12/21   Signature : Bazgha Razi

Optimal binary search tree



For left subtree
[i, k-1]
and for right subtree
[k, j]

∴ Optimal binary search tree of minimum cost 32 is