Name: Vishal Pandey
Class :BCA-B
Roll No:41221191
Subject:DAA
Date: 7 june 2023

(1)

Q1 Write Recursive divide and conquer algorithm to multiply large integer numbers and show the steps for multiplication of 2101 * 1130

- $a = a_1 a_0$ and $b = b_1 b_0$
- $c = a * b$

$= (a_1 * b_1) 10^n + (a_1 * b_0 + a_0 * b_1) 10^{n/2} + (a_0 * b_0) C_2 10^n + C_1 10^{n/2} + C_0$

where $C_2 \times 10^4 + C_1 \times 10^2 + C_0 \times 10^0$

$C_2 = a_1 * b_1$ is the product of their first halves

$C_0 = a_0 * b_0$ is the product of their second halves

$C_1 = (a_1 + a_0) * (b_1 + b_0) - (C_2 + C_0)$ is the product of the sum of the a's halves and sum of the b's halves minus the sum of C_2 and C_0.

$M(n) = 3M(n/2)$ for $n > 1$, $M(1) = 1$

$= 3M[3M(n/4)]$

$= 3^3 M[n/8$

$= 3^k M[n/2^k]$

$a = 2101$ $\quad (a_1 = 21 \quad a_0 = 01)$

$b = 1130$ $\quad (b_1 = 11 \quad b_0 = 30)$

$C_2 = a_1 * b_1 = 21 * 11 = 231$

$C_0 = a_0 * b_0 = 1 * 30 = 30$

$C_1 = (a_1 + a_0) * (b_1 + b_0) - (C_2 + C_0)$

$= (21 + 01) * (11 + 30) - (231 + 30)$

$= 902 - 261$

$= 641$

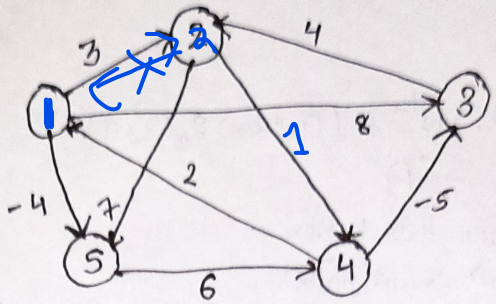
$C_2 \times 10^4 + C_1 * 10^2 + C_0 \times 10^0$

$= 231 \times 10^4 + 30 \times 10^2 + 30$

$= 2374130$

Name: Vishal Pandey
Class :BCA-B
Roll No:41221191
Subject:DAA
Date: 7 june 2023

Q2 Consider the following graph and apply floyd warshall algorithm



$A^k[i,j] = \min(A^{k-1}[i,j], A^{k-1}[i,k] + A^{k-1}[k,j])$

$D^{(0)} =$

0	3	8	∞	-4
∞	0	∞	1	7
∞	4	0	∞	∞
2	∞	-5	0	∞
∞	∞	∞	6	0

$D^{(1)} =$

0	3	8	∞	-4
∞	0	∞	1	7
∞	4	0	∞	∞
2	5	-5	0	-2
∞	∞	∞	6	0

$D^{(2)} =$

0	3	8	4	-4
∞	0	∞	1	7
∞	4	0	5	11
2	-1	-5	0	-2
∞	∞	∞	6	0

$D^{(3)} =$

0	3	8	4	-4
∞	0	∞	1	7
∞	4	0	5	11
2	-1	-5	0	-2
∞	∞	∞	6	0

$D^{(4)} =$

0	3	-1	4	-4
3	0	-4	1	-1
7	4	0	5	3
2	-1	-5	0	-2
8	5	1	6	0

$D^{(5)} =$

0	1	-3	2	-4
3	0	-4	1	-1
7	4	0	5	3
2	-1	-5	0	-2
8	5	1	6	0

Name: Vishal Pandey
Class :BCA-B
Roll No:41221191
Subject:DAA
Date: 7 june 2023

(3)

3. write strassen's algorithm for matrix multiplication problem. show the execution of algorithm with sample example.

$$X\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times Y\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = C\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$P = (A_{11} + A_{22})(B_{11} + B_{22})$

$Q = (A_{21} + A_{22})B_{11}$

$R = A_{11}(B_{12} - B_{22})$

$S = A_{22}(B_{21} - B_{11})$

$T = (A_{11} + A_{12})B_{22}$

$U = (A_{21} - A_{11})(B_{11} + B_{12})$

$V = (A_{12} - A_{22})(B_{21} + B_{22})$

$C_{11} = P + S - T + V$

$C_{12} = R + T$

$C_{21} = Q + S$

$C_{22} = P + R - Q + U$

Example

$$A = \left[\begin{array}{cc|cc} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ \hline 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{array}\right] \qquad B = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ \hline 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{array}\right]$$

$A_{11} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}$ $A_{12} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$ $B_{11} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}$ $B_{12} = \begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}$

$A_{21} = \begin{bmatrix} 0 & 1 \\ 5 & 0 \end{bmatrix}$ $A_{22} = \begin{bmatrix} 2 & 1 \\ 3 & 1 \end{bmatrix}$ $B_{21} = \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix}$ $B_{22} = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$

$P = \begin{bmatrix} 4 & 0 \\ 6 & 2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 7 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix}$

$Q = \begin{bmatrix} 3 & 1 \\ 7 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix}$

Name: Vishal Pandey
Class :BCA-B
Roll No:41221191
Subject:DAA
Date: 7 june 2023

(4)

$$R = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ -5 & 4 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix}$$

$$S = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 3 & 1 \\ 5 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix}$$

$$U = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -2 & -1 \end{bmatrix}$$

$$V = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix}$$

$$C_{11} = P + S - T + V$$

$$\begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix} + \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix} + \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

$$C_{12} = R + T$$

$$\begin{bmatrix} -1 & 0 \\ 9 & 4 \end{bmatrix} + \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ 1 & 9 \end{bmatrix}$$

$$C_{21} = Q + S$$

$$\begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix} + \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 1 \\ 5 & 8 \end{bmatrix}$$

$$C_{22} = P + R - Q + U$$

$$\begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix} - \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix} + \begin{bmatrix} 2 & 3 \\ -2 & -3 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 7 \\ 7 & 7 \end{bmatrix}$$

Name: Vishal Pandey
Class :BCA-B
Roll No:41221191
Subject:DAA
Date: 7 june 2023

Q4 Differentiate between Dynamic Programming and Divide and conquer strategy.

Divide and Conquer method	Dynamic Programming
1. It deals three steps at each level of recursion: Divide the problem into subproblems : conquer the subproblem by solving them recursively : Combine the solutions to the subproblem into the sol" for original subproblems	1. It involves the sequence of four steps • Characterize the structure of optimal sol" • Recursively defines the values of optimal sol" • Compute the value of optimal sol" in a Bottom up minimum • Construct an optimal sol" from computed information
2. It is Recursive	2. It is non Recursive.
3. It is a top down approach	3. It is a Bottom-up approach
4. For ex: merge sort & Binary search etc	4. For ex: matrix multiplication

Q5 Differentiate between P, NP, NP-C and NP-Hard problems with a suitable diagram

1. P (Polynomial Time): Problems in this class can be solved by an algorithm that runs in polynomial time. In other words, the running time of the algorithm is bounded by polynomial function of the input size. Example include sorting, searching, and basic arithmetic operations.

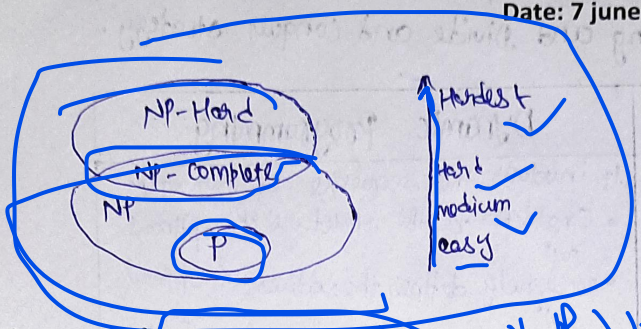
2. NP (Nondeterministic Polynomial): Problems in this class have a non-deterministic algorithm that can verify a potential solution in polynomial time. However, finding an actual sol" may require an exponential amount of time. NP problems are considered "hard" to solve directly but can be verified efficiently. Examples travelling salesman problem and the knapsack problem.

3. NP-Complete (NPC): A problem is NP-complete if every problem in the NP class can be reduced to it in polynomial time. If a polynomial time algorithm can be found for any NP-complete problem, it implies that P = NP, which is an unsolved question in computer science. Examples Boolean satisfiability problem and Traveling salesman problem.

4. NP-hard: This class includes problems that are at least as hard as the hardest problems in NP but may not be in NP themselves. NP hard problems do not necessarily have to be verifiable in polynomial time, but they have a polynomial time reduction from an NP-complete problem. Example Hamiltonian cycle problem.

Name: Vishal Pandey
Class :BCA-B
Roll No:41221191
Subject:DAA
Date: 7 june 2023

6



NP Hard

Q6 What is Halting problem?

The Halting problem is the problem of deciding or concluding based on a given arbitrary computer program and its input, whether that program will stop executing or run in an infinite loop for the given input.

The Halting problem tells that it is not easy to write a computer program that executes in the limited time that is capable of deciding whether a program halts for an input.

Q7 List the problems belonging to polynomial class

$T(n) = O(C * n^k)$ where $C > 0$ and $k > 0$ where C and K are constants and n is input size

- All basic mathematical operations; addition, subtraction, division, multiplication
- Testing for primary
- Hashtable lookup, storing operations, sorting problems
- shortest path Algorithms
  Dijkstra, Bellman-ford, floyd-warshall
- Linear and Binary search Algorithms for a given set of numbers.