Muhammad Nadeem
2019800350

Analysis of Algorithms
CS-318-A

Assignment # 3

Q.No1:- What is the difference between Divide and conquer, Dynamic Programing and Memoization?

Divide and Conquer:-

It is a recursive technique that involves breaking a large and complex problems into simpler and smaller subproblems. The solutions of these subproblems are combined to solve the original problem.

Dynamic Programming:-

Dynamic Programming is a non-recursive optimization technique that involves breaking a large problem into subproblems. The result of these subproblems are solved to avoid redundant calculations:-

Memoization:-

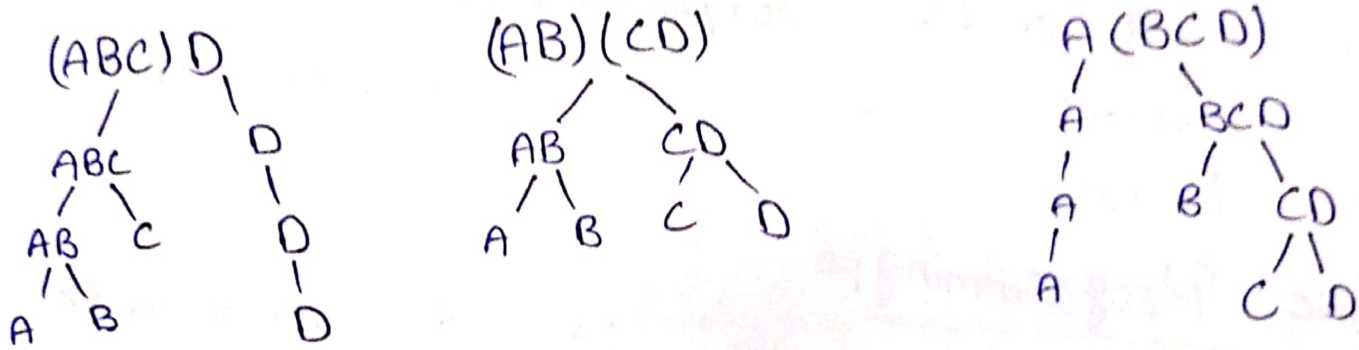
Memoization is a technique used in Dynamic Programming. Its use is to optimize the computation of subproblems by storing their solution in a memoization table or an array.

Matrix chain Multiplication using Memoization:-

Memoization has improved Matrix chain Multiplication is such a way that it reduces calculations by storing the values of each subproblem in a memoization table of size n.

Example:-

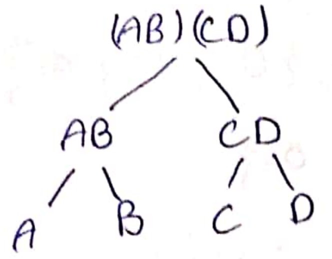
Assume we have four matrices A, B, C, D then we get many combinations. Assuming 3 of the combinations we get the recursive trees.



In these trees we get repeating multiplication Combinations.

We needed a method where, Somehow we are are able to store the results of subproblems. This can be ~~store~~ done through Memoization.

so Suppose we solve the tree/Combination :



The values of each subproblem is solved in the table and when we solve another tree, before calculating the costs, we check, if the value of that subproblem is already calculated. If it is, we simply use that value. proof of correctness of activity selection problem can be passed in two parts

i) Problem shows greedy choice

ii) Problem's optimal solution is made up of optimal solution of sub problems

i) Property of greedy choice:-

Suppose Greedy Solution = $G = \{a_1, a_2, \dots a_k\}$

Suppose actual optimal solution = $O = \{o_1, o_2 \dots o_y\}$

Here Greedy solution is constructed by early finish time.

Optimal solution may or may not be sorted. Also, suppose first activity of optimal solution is not equal to the first activity of greedy solution.

$a_1 \neq o_1$

Intermediate set $I = O - \{o_1\} \cup \{a_1\}$

$I = \{a_1, o_2, \ldots\ldots, o_n\}$

Is I feasible or optimal?

If I is feasible and optimal both then activity 1 (a_1) will be a part of optimal solution Since a_1 has earliest finish time so it is feasible

Since activities in I is same as in O so, it is optimal solution.

ii) Optimal substructure:-

→ Suppose S is the activity selection Problem and O is the optimal solution

→ $O' = O - \{a_1\}$

$S' = \{i \in S \mid S_i > f_1\}$

Starting time should be greater than finishing time.

→ O' is the optimal solution for S'.

→ Suppose O' is not the optimal solution of S' and solution I exists for S' with more activities.

$A = I \cup \{a_1\}$

→ Since O is the optimal solution for S, so, it is not possible.

Hence, O' must be optimal solution for S'.