Utilization Of Algorithms, Dynamic Programming, Optimal Memory Utilization

TEAM ID	NM2023TMID04391
PROJECT TITTLE	BLOCKCHAIN POWERED LIBRARY MANAGEMENT

INTRODUCTION:

- Dynamic programming is a technique that breaks the problems into sub-problems, and saves the result for future purposes so that we do not need to compute the result again.
- The subproblems are optimized to optimize the overall solution is known as optimal substructure property.
- The main use of dynamic programming is to solve optimization problems.
- Here, optimization problems mean that when we are trying to find out the minimum or the maximum solution of a problem.
- The dynamic programming guarantees to find the optimal solution of a problem if the solution exists.

Approaches of dynamic programming

- There are two approaches to dynamic programming:

- Top-down approach

- Bottom-up approach
- Top-down approach

- The top-down approach follows the memorization technique, while bottom-up approach follows the tabulation method.
- Here memorization is equal to the sum of recursion and caching.
- Recursion means calling the function itself, while caching means storing the intermediate results.

Advantages

- It is very easy to understand and implement.
- It solves the subproblems only when it is required.
- It is easy to debug.

Disadvantages

- It uses the recursion technique that occupies more memory in the call stack.
- Sometimes when the recursion is too deep, the stack overflow condition will occur.

- It occupies more memory that degrades the overall performance.

Let's understand dynamic programming through an example.

```
int fib(int n)
{
   if(n<0)
   error;
 if(n==0)
 return 0;
 if(n==1)
return 1;
sum = fib(n-1) + fib(n-2);
```