

1) What is memoization in dynamic programming?

- a) A technique to optimize space complexity
- b) A technique to optimize time complexity
- c) A technique to store and reuse already computed results
- d) A technique to minimize the number of subproblems

Explanation: ডাইনামিক প্রোগ্রামিং এ memoization এর মাধ্যমে আমরা পূর্বে কোনো কিছুর ভ্যালু বের করা থাকলে তা পরবর্তীতে নতুন ভাবে বের না করে, আগের ভ্যালু টি ব্যবহার করে থাকি, এ ক্ষেত্রে প্রত্যেক instance এর ভ্যালু গুলো আমরা array তে স্টোর করে থাকি, একে memoization বলা হয়।

2) What is the time complexity of the naive recursive approach to compute the nth Fibonacci number?

- a) $O(n)$
- b) $O(2^n)$
- c) $O(n^2)$
- d) $O(\log n)$

Explanation: যেহেতু প্রত্যেকটি fibonacci বের করার জন্য এর আগের দুটি fibonacci বের করতে হয়, অর্থাৎ প্রত্যেকটি ফাংশন কল থেকে দুটি করে ফাংশন কল হয়, তাই n'th fibonacci এর জন্য total function কল হয় প্রায় 2^n টি। তাই এর টাইম কমপ্লেক্সিটি $O(2^n)$ । বিস্তারিত মডিউলে ব্যাখ্যা করা আছে।

3) How does dynamic programming improve the time complexity of solving problems with overlapping subproblems?

- a) By avoiding recursion
- b) By using a greedy approach
- c) By memoization
- d) By increasing the input size

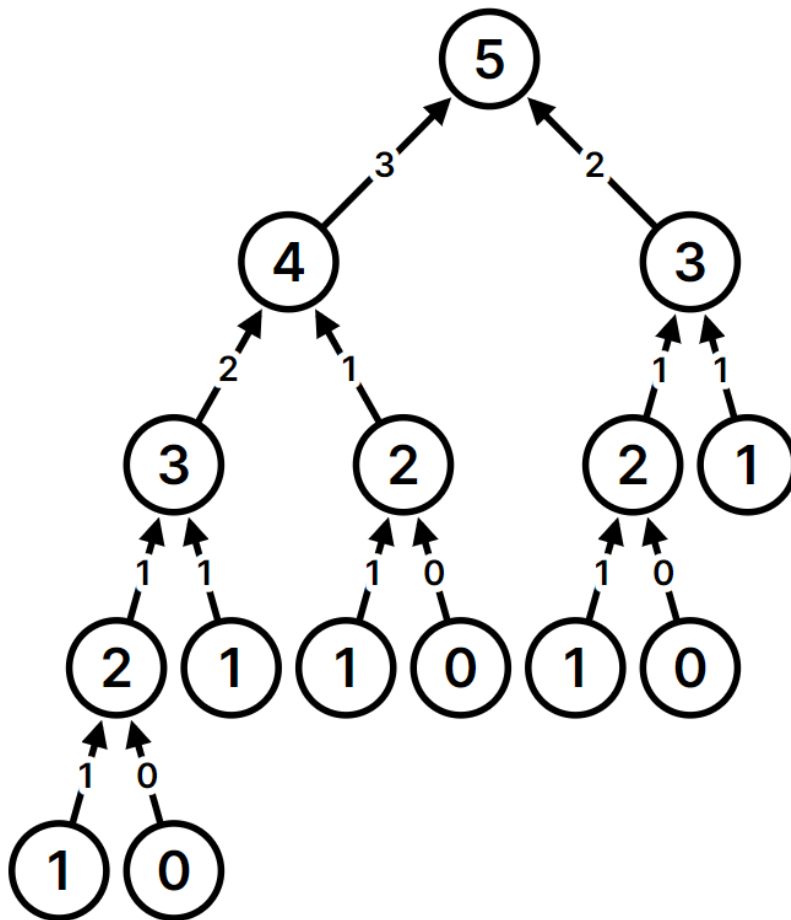
Explanation: ডাইনামিক প্রোগ্রামিং এ memoization এর মাধ্যমে আমরা পূর্বে কোনো কিছুর ভ্যালু বের করা থাকলে তা পরবর্তীতে নতুন ভাবে বের না করে, আগের ভ্যালু টি ব্যবহার করে থাকি। এ ক্ষেত্রে overlapping instance গুলো বারবার কল হয় না, তাই টাইম কমপ্লেক্সিটি $O(2^N)$ থেকে কমে $O(N)$ এ চলে আসে।

4. What is the main problem of using a naive recursive approach to compute Fibonacci numbers?

- a) It uses too much space
- b) It has a high time complexity due to repeated calculations
- c) It requires sorting the sequence first

Explanation: যেহেতু Recursion এর সাহায্যে, প্রত্যেকটি fibonacci বের করার জন্য এর আগের দুটি fibonacci বের করতে হয়, অর্থাৎ প্রত্যেকটি ফাংশন কল থেকে দুটি করে ফাংশন কল হয়, তাই n'th fibonacci এর জন্য total function কল হয় প্রায় 2^n টি। যা খুবই ধীরগতির হয়ে থাকে,

5)



What will be the suitable recursive function for this recursion tree?(lets say base case is if $n \leq 1$ return n)

- a) $fn(n-1)*n$
- b) $fn(n-1)+fn(n-2)$
- c) $fn(n-1)+n$
- d) $fn(n-1)/n$

Explanation: উক্ত Recursive function এর tree টি মূলত মডিউলে দেখানো fibonacci বের করার recursive tree টি।

6) What is the time complexity of calculating the factorial of a number N ?

- a) $O(N)$
- b) $O(N*\log n)$
- c) $O(2^N)$
- d) $O(N^2)$

Explanation: N এর ফ্যাক্টরিয়াল বের করার Time complexity হলো $O(N)$ যা মডিউলে বিস্তারিত বলা আছে।

7 What is the issue with the following recursive Fibonacci function?

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

- a) Incorrect base condition
- b) Higher time complexity due to repeated calculations of same value**
- c) It does not return the correct Fibonacci number
- d) It will result in a segmentation fault

Explanation: উক্ত কোড টি মূলত মডিউলে দেখানো naive way তে fibonacci বের করার recursive কোড। উক্ত কোডে প্রত্যেকটি fibonacci বের করার জন্য এর আগের দুটি fibonacci বের করতে কল করা হয়, অর্থাৎ প্রত্যেকটি ফাংশন কল থেকে দুটি করে ফাংশন কল হয়, তাই n'th fibonacci এর জন্য total function কল হয় প্রায় 2^n টি. যা খুবই ধীরগতির হয়ে থাকে।

8. In a dynamic programming approach using an array to store Fibonacci numbers, at which index will the Fibonacci value 5 be stored?

- a. 3
- b. 2
- c. 4
- d. 5

Explanation: মডিউলে দেখানো dp array এর মধ্যে dp[5] এ fibonacci(5) এর ভ্যালু রাখা হয়েছে।

9. In the module, how did we check if we already calculated the value of the nth fibonacci number?

- a. dp[i] != -1

B. $dp[i] = -1$

C. $dp[n] != -1$

D. $dp[n] = -1$

Explanation: মডিউলে দেখানো dp array এর মধ্যে -1 থাকলে আমরা বুঝতে পারি, উক্ত state বা ফাংশনটির জন্য এখনো আমরা উত্তর পাইনি বা এখনো সম্ভব করা হয় নি। সেফ্রে আমরা পরবর্তী কাজ গুলো করে থাকি।

10. What is the space complexity of calculating fibonacci using bottom up approach ?

a. $O(2^n)$

b. $O(\log N)$

c. $O(N)$

d. $O(1)$

Explanation : Bottom up approach এর ক্ষেত্রে আমরা একটি N সাইজের array নিয়ে সেখানে fibonacci গুলো ক্যালকুলেট করে থাকি।