Day 8

02 December 2022 21:06

$F(0) = 0$
$F(1) = 1$

$F(2) = F(1) + F(0)$

$F(2) = 1$

$F(3) = F(2) + F(1)$

$= 1 + 1 = 2$

$F(4) = F(3) + F(2)$

$n = 2$

$F(2) = F(1) + F(0)$

$= 1 + 0 = 1$

$F(3) = F(2) + F(1)$

$= 1 + 1 = 2$

$fibo(n) = fibo(n-1) + fibo(n-2)$

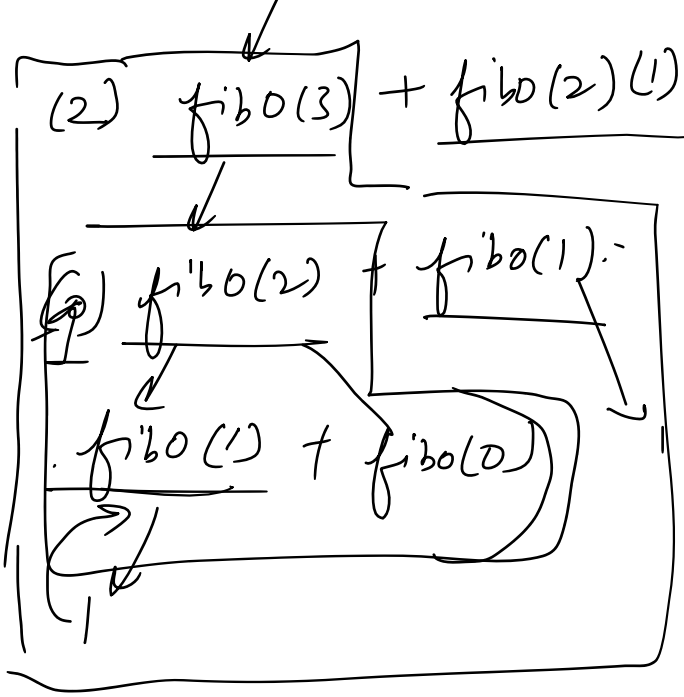
```
int fibo(int n){
    if(n==0) return 0; if (n==1) return 1;
    return fibo(n-1) + fibo(n-2);
}
```

Input n = 2000

n = 5

fibo(5)

(3) fibo(4) + fibo(3) (recalculation)



n = 2000

fib(2000)

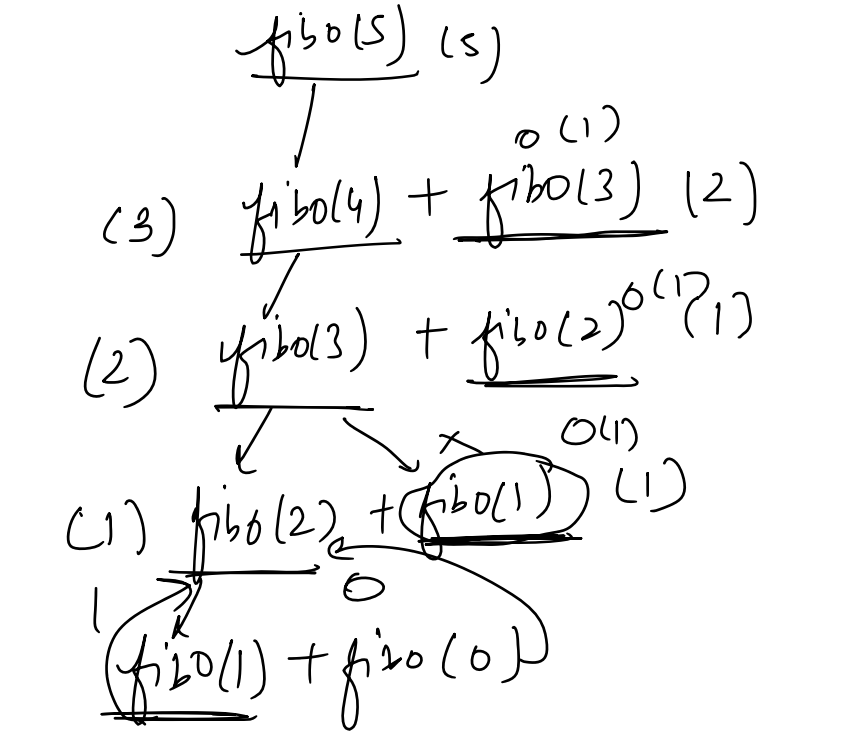
fib(1999) + fib(1998) (recalculate)

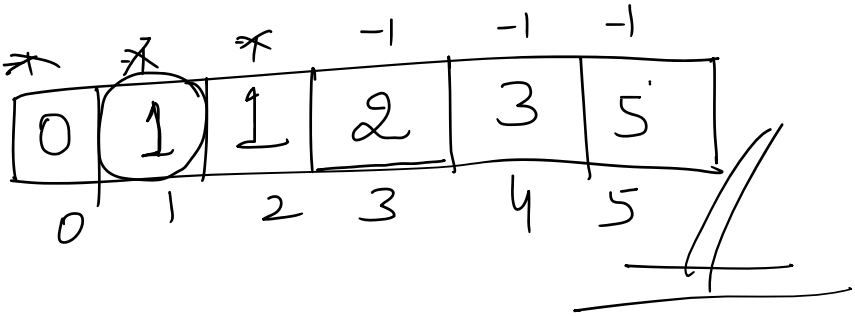
fib(1998)

fib(1998)

Dynamic Programming

n=5



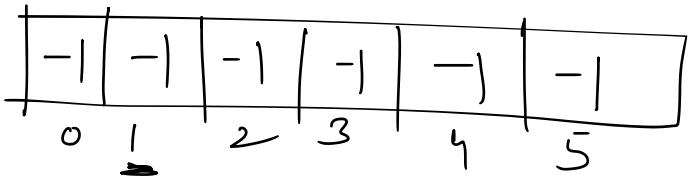


☆ Rcursive calls +1 Answer Array

में store करता जाएगा.

Why?

ताकि Recalculate ना करना पड़े



-1 signifies calculation नहीं हुई है।
-1 नहीं है तो calculation हो चुकी है।

Recursion — O(2^N)

D.P — O(N)

$2^N \Rightarrow O(N)$

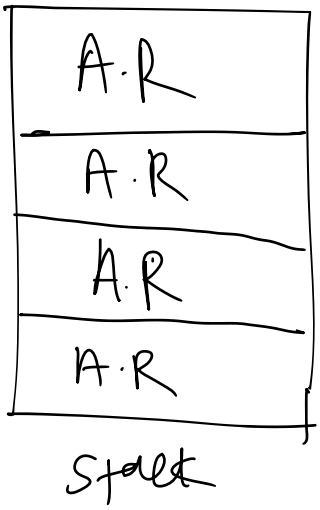
$2^{1000} \Rightarrow O(2000)$

Recursion + Auxillary Space

Memoization

Tabulation (loop)

① Recursion (Top to Bottom)



Buffer Overflow

Recursive function 80 %

Auxiliary

② Loops

(Tabulation)

(Bottom to Top)

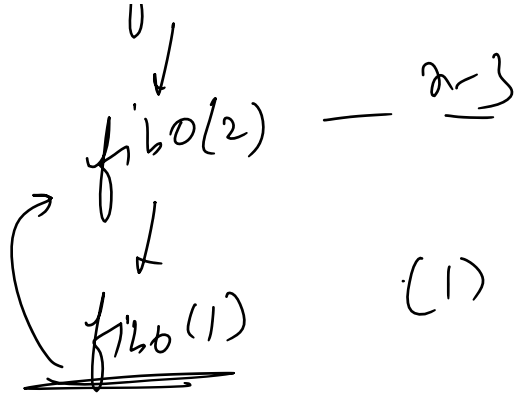
Recursion Top to Bottom

fibo(5) nth

fibo(4) — n-1

fibo(3) — n-2

n-3

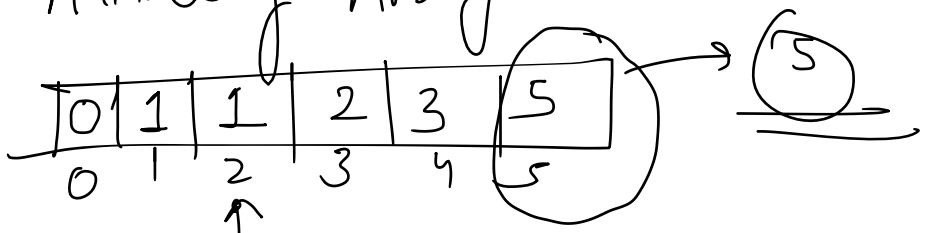


Tabulation

Bar Case to n^{th} value

$n=5$

(1) Auxillary Array



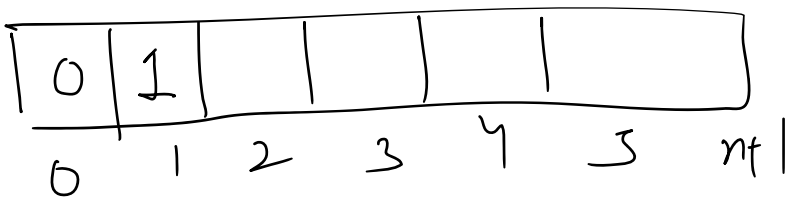
$f(2) = f(1) + f(0)$

$f(3) = f(2) + f(1)$

$f(4) = f(3) + f(2)$

$f(5) = f(4) + f(3)$

(1)



(2) Bar Case Value Auxillary Array

$f(n) = f(n-1) + f(n-2)$

$$f(n) = f(n-1) + f(n-2)$$

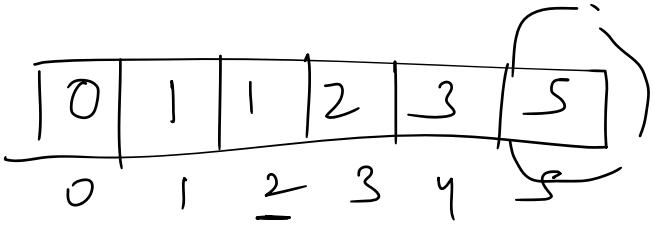
Recursive

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

dp[n-1] + dp[n-2]

T.C – O(N)

S.C – O(N)




```
for(int i=2; i<=n; i++){
  dp[i] = dp[i-1]
        + dp[i-2];
}
return dp[n];
```

T.C – O(N)

S.C – O(1)

N-th Tribonacci Number

1137. N-th Tribonacci Number Hint

Easy 2.4K 132

Companies

The Tribonacci sequence T_n is defined as follows:

$T_0 = 0$, $T_1 = 1$, $T_2 = 1$, and $T_{n+3} = T_n + T_{n+1} + T_{n+2}$ for n >= 0.

Given `n`, return the value of T_n.

Example 1:

1137. N-th Tribonacci Number

Hint

Easy 2.4K 132

Companies

The Tribonacci sequence T_n is defined as follows:

$T_0 = 0, T_1 = 1, T_2 = 1$, and $T_{n+3} = T_n + T_{n+1} + T_{n+2}$ for $n >= 0$.

Given `n`, return the value of T_n.

Example 1:

```
Input: n = 4
Output: 4
Explanation:
T_3 = 0 + 1 + 1 = 2
T_4 = 1 + 1 + 2 = 4
```

Example 2:

```
Input: n = 25
Output: 1389537
```

$n=4$

$T_{n+3} = T_1 + T_2 + T_3$

$T_3 = T_0 + T_1 + T_2$

$T_0 = 0 \quad T_1 = 1 \quad T_2 = 1$ // Base

$T_n = T_{n-3} + T_{n-2} + T_{n-1}$ // Recursive

$T_4 = T_1 + T_2 + T_3$

Recursion - (80 %) → Memoization (10 %)

Recursion (80%) → Memoization ✓

↓ (10%)

Tabulation (bottom to top)

Recursive Code

```
int Tribo(int n){
  if(n==0) return 0; if(n==1 || n==2) return 1;
  return Tribo(n-3) + Tribo(n-2) + Tribo(n-1);
}
```

T.C - O(3^N)

S.C - O(N)

Time Limit Exceeded