First n Fibonacci

Created by Chandra Prakash

Problem Statement

Given a number **n,** return an array containing the first **n** Fibonacci numbers.Note: The first two numbers of the series are 0 and 1.

Examples:

```
Input:n = 5
Output:[0, 1, 1, 2, 3]
```

```
Input:n = 7
Output:[0, 1, 1, 2, 3, 5, 8]
```

```
Input: n = 2
Output: [0, 1]
```

Constraints: 1 <= n <= 30

Editorial

Approach

To generate the first `n` Fibonacci numbers efficiently, we can use **Dynamic Programming (DP)**. The idea is to build the sequence iteratively and store the previously computed values to avoid redundant calculations.

We initialize a DP array `fib` of size `n`, where:

- `fib[0] = 0`
- `fib[1] = 1` (if `n > 1`)

From index 2 to `n-1`, we compute:

```
fib[i] = fib[i-1] + fib[i-2]
```

Code

```
#include <bits/stdc++.h>
using namespace std;

class Solution {
public:
    vector<int> fibonacci(int n) {
        vector<int> fib(n);
        fib[0] = 0;
        if (n > 1) fib[1] = 1;
```

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

int main() {
    int n;
    cin >> n;
    Solution s1;
    vector<int> result = s1.fibonacci(n);
    for (int num : result) cout << num << " ";
    return 0;
}
```

Complexity Analysis

- **Time Complexity:** `O(n)` — Each Fibonacci number from index 2 to `n-1` is computed once.
- **Space Complexity:** `O(n)` — We use a vector of size `n` to store the sequence.
