Intro to DP

Steps to Solve any DP Problem:→

① form a state → f(x) ← one Dimention
→ f(x,y) ← two Dimention

② form a recursive relation.
↳ Fibonacci: F(n) = f(n-1) + f(n-2)
state ↳ Recursive

③ find the optimal Solution.
Tabulation ← → Memoization

" Tabulation VS Memoization"

memoization means that save the value in the array at the first call, then if I need this value instead of calling it again Recursivly, no Just Pick this value and stop recursive call for this part.

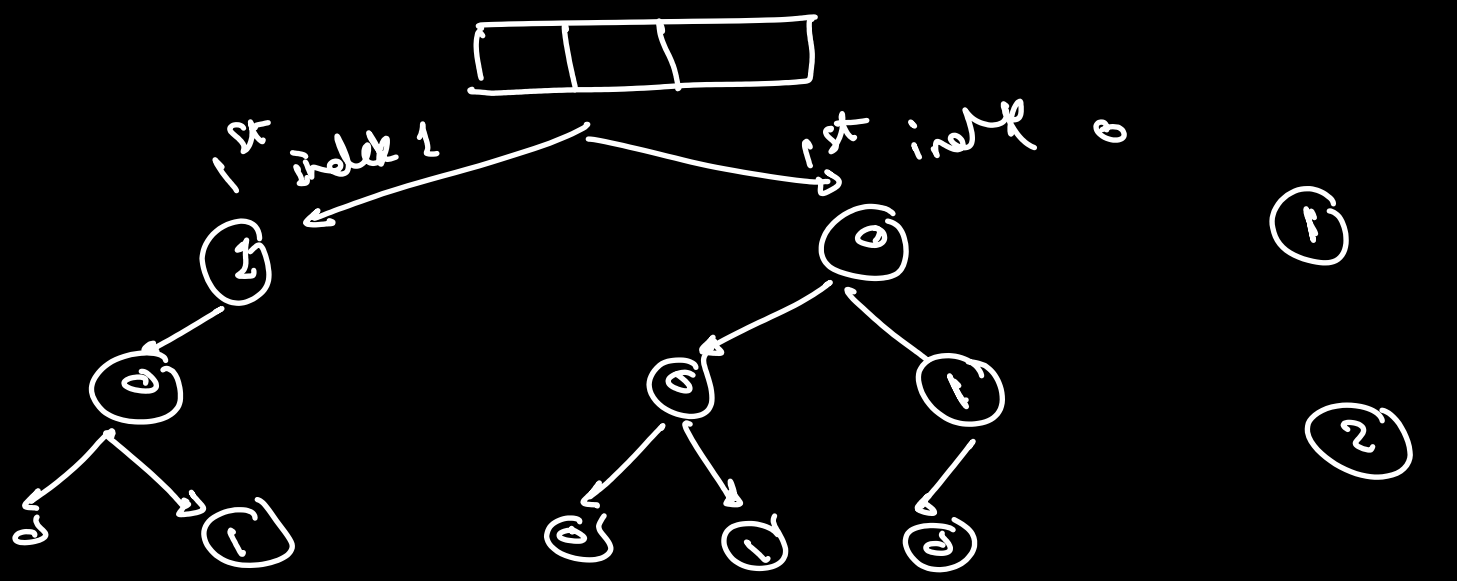
```cpp
int calcFib(int n) {
    // Base Case
    if(n <= 2) return 1;
    if(memo[n] != -1) return memo[n];
    // Recursive Case
    return memo[n] = calcFib(n-1) + calcFib(n-2);
}
int main() {
    int n;
    cin >> n;
    memo.resize(n + 1, -1);
    cout << calcFib(n);

}
```

tablation:→ means to do the same approach but in recursive approach, and fill the Base Cases Manually Before Start iteration.

```cpp
int main() {
    int n;
    cin >> n;
    vector<int> fib(n + 1);
    fib[1] = fib[2] = 1;
    for(int i = 3; i <= n ; i++) {
        fib[i] = fib[i - 1 ] + fib[i - 2];
    }
    cout << fib[n];
}
```

* No of ways to form atree without to Consecutive ones



Solution ⇒

0 0 0
0 0 1
0 1 0
1 0 0
1 0 1

5 States.