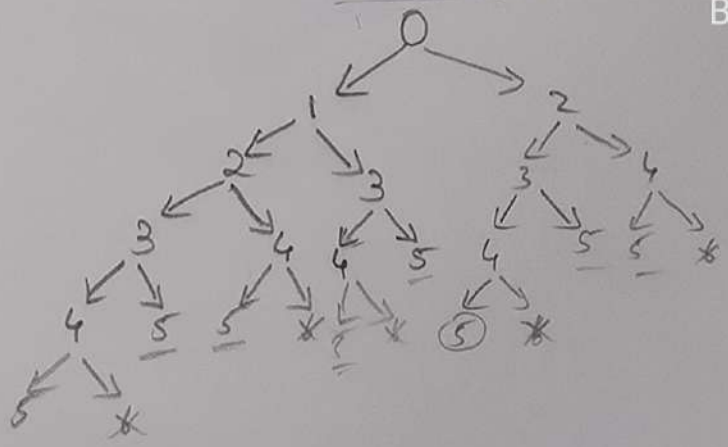
Suppose n=5

Decision Tree



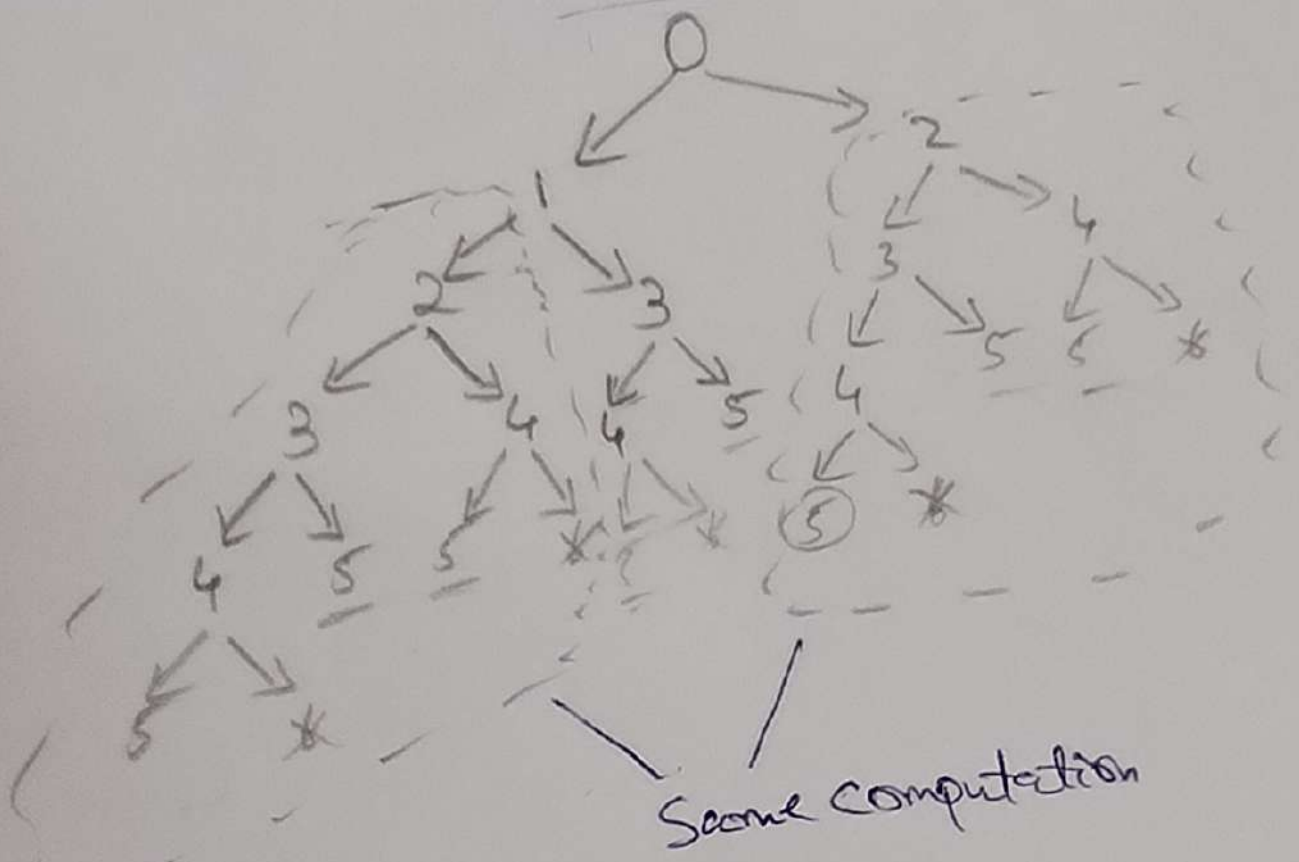
Brute Force

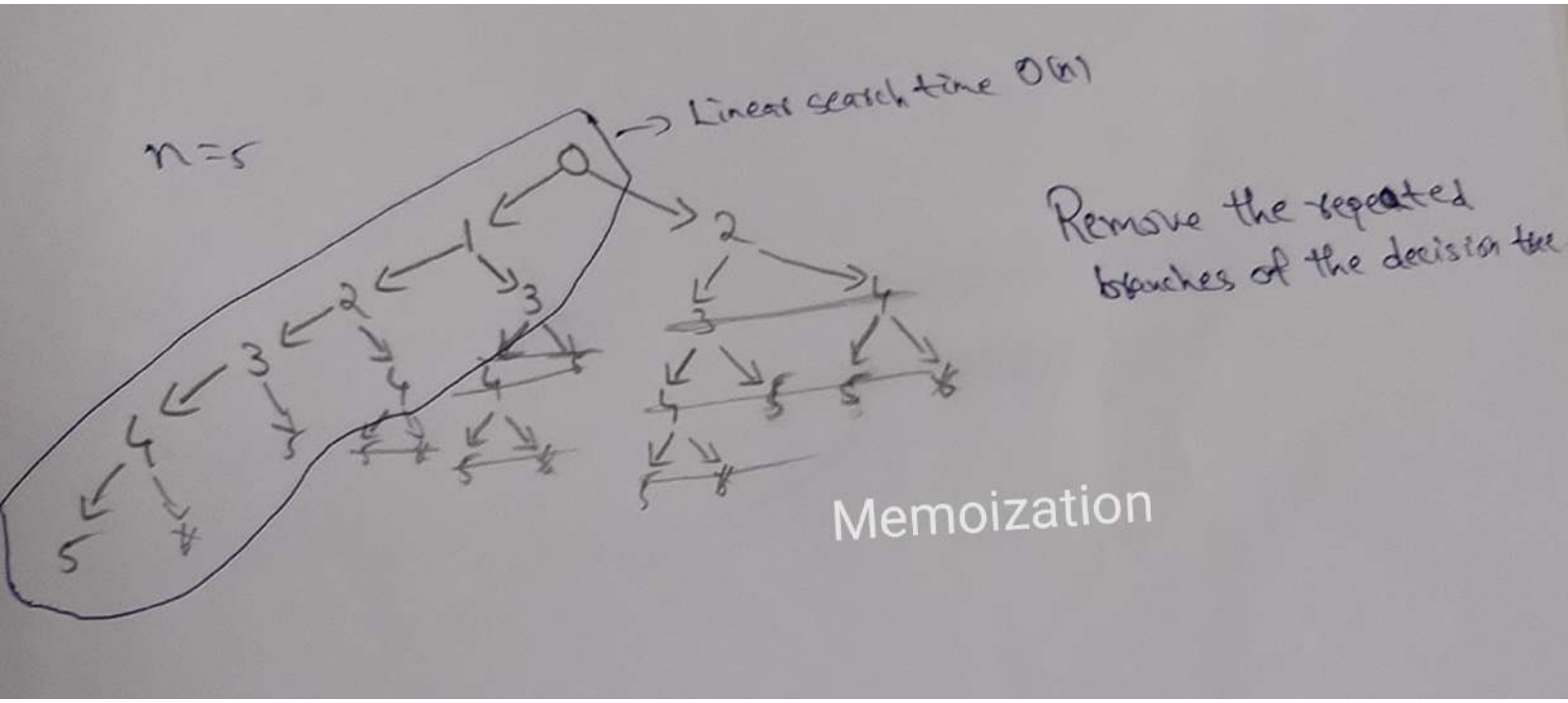
$T(n) = O(2^n)$ = Time complexity

1)

Suppose n = 5

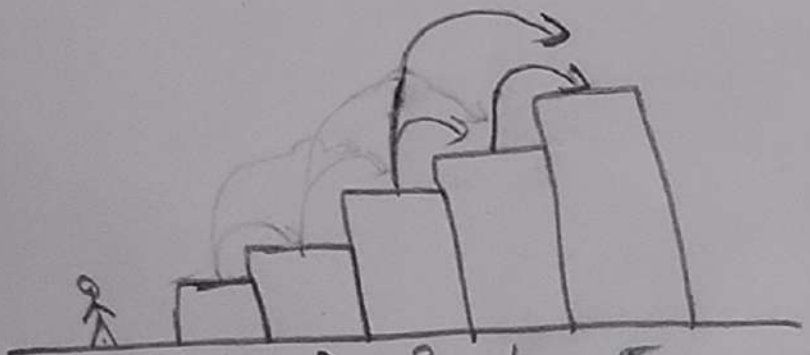
Decision Tree



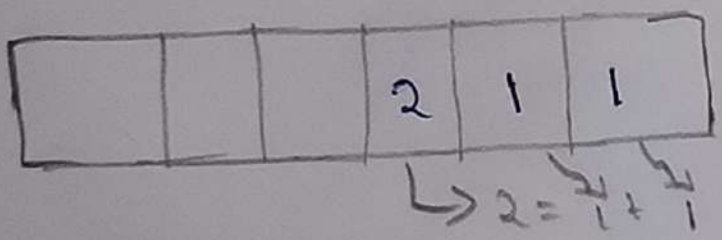

n=5
Linear search time O(n)
Remove the repeated
branches of the decision tree
Memoization

DP - Bottom up

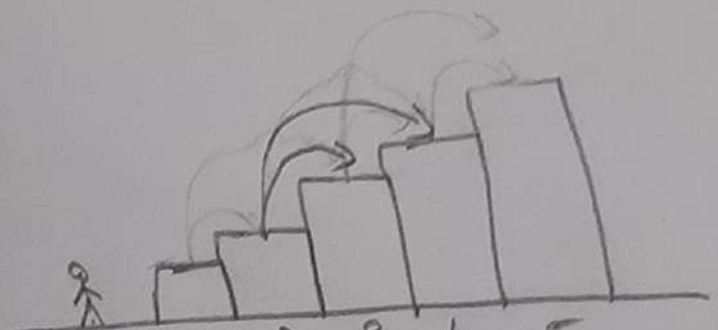
Start at base case and work the way up



DP



↳ 2 = 1 + 1



DP

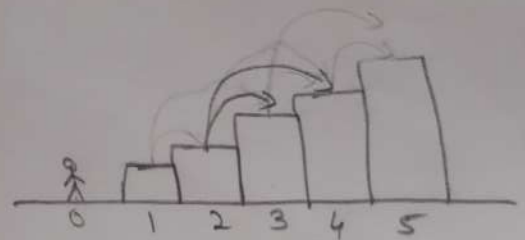
8	5	3	2	1	1

B = 5 + 3 (and so on)

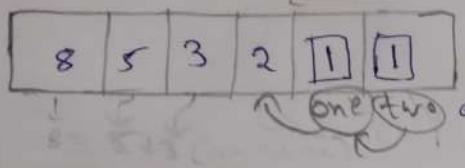
Fibonacci sequence (in reverse ←)

→ cache them

DP – Bottom up
(start at base case and work your way up)



DP



fibonacci sequence (in reverse ⇐)

⇐ To avoid using extra memory we'll initialize two variables to compute the next number in the array.

Python3 • Auto

```python
class Solution:
    def climbStairs(self, n: int) -> int:
        one, two = 1, 1

        for i in range(n-1):
            temp = one
            one = one+two
            two= temp

        return one
```