11. Dynamic Programming by Yan Chao Feng

Introduction

Suppose you can climb a set of stairs by either taking 1, 3, or 4 stairs at once per step. Compute the number of ways to climb 150 stairs using dynamic programming. Dynamic programming is a method that breaks down a complex problem into smaller problems and stores the solutions of these smaller problems to solve the entire problem. We need to use a data structure that would allow us to manipulate and store those solutions to avoid repeated calculations and reuse them again, such as a list. We will be deep diving on how does the algorithm actually work and soon discover how many different ways can someone climb 150 stairs.

Analysis

```
def dynamic_step(n):

    if n <= 4:
        step_count = [0, 1, 1, 2, 4]
        return step_count[n]

    else:
        step_count = [0, 1, 1, 2, 4] + [0 for i in range(n-4)]

        for k in range(5, n+1):
            step_count[k] = step_count[k-1] +   step_count[k-3] +   step_count[k-4]
        return step_count[n]

print(dynamic_step(150))
```

```
1167216219038060952672162558384 9
```

Let's break this program down into something everyone understand.

The function, called **dynamic_step**, accepts a single n parameter that represents stairs. It should accept any positive integer $n >= 0$.

Inside the body of the program, it contains two conditions. If it satisfies $n <= 4$ statement, then it should return the result of number ways for either 0, 1, 2, 3, or 4 stairs depending on n.

0	1	1	2	4
0	1	2	3	4

Now, the secret to solving the number of ways to climb n is to look backward and the table shown above is the only data needed to do the rest of the calculations.

Bottom row represent stairs and the top row corresponds the number of ways to get there.

The first value 0 signifies the starting point, or perhaps someone who decided not to climb the stairs.

To reach 1 stair, it is only possible to move 1 stair. Thus, there is only 1 way.

To reach 2 stair, it is only possible to move 1 stair at a time to get to 2 stair. Thus, there is only 1 way.

To reach 3 stair, you can either move 1 stair at a time or you can move 3 stairs. Thus, there are 2 possible ways.

To reach 4 stair, you can either move 1 stair at a time, move 3 stair from 1 stair, move 1 stair from 3 stair, or choose to move 4 stair at once. Thus, there are 4 possible ways.

It seems simple enough, but it will get fairly complicated and confusing as it progresses to higher numbers. Luckily, there is a better way to find the number of ways for n stair almost easily by only using the values from the table as represented by this pseudocode.

Number of Steps of *n* Stair = Number of Steps from *n-1* Stair + Number of Steps from *n-3* Stair + Number of Steps from *n-4* Stair

If we substitute n with 5, we get this statement.

Number of Steps of *5* Stair = Number of Steps from *4* Stair + Number of Steps from *2* Stair + Number of Steps from *1* Stair

Number of Steps of *5* Stair = 4 + 1 + 1 = 6

Figuring out the number of ways for a stair relies on using values from previous stairs. Hence, we are technically looking backward and that is why it is important to save our results to do this same type calculation. So, we must include the data from the table to our algorithm to

make the entire thing work. And that is essentially the logic behind our else condition if n is not less than or equal to 4.

If we pass 150 to the function, it should satisfies the else condition.

It creates a list that contains 151 item of zeroes because we want to match stair to their identical index.

[0 (index 0) ,0,0, ..., 0 (index 150)]

Then manually assign values of the table to the corresponding index.

Afterward, the for loop basically does the same type of calculation repeatedly just as the pseudocode demonstrated. It will keep performing the calculations needed to find the number of ways to climb 150 stairs using dynamic programming.

Conclusion

We finally have a implemented an algorithm to the find the number of ways to climb 150 stairs by either taking 1, 3, or 4 stairs at once per step. The final result is an astounding **11,672,162,190,380,609,526,721,625,583,849** possible ways to climb 150 stairs.