Those who cannot remember the past, are condemned to repeat it.

*- Dynamic Programming*

The quote above says a lot about Dynamic Programming. So, is repeating the things for which you already have the answer, a good thing ? A programmer would disagree. That's what Dynamic Programming is about. To always remember answers to the sub-problems you've already solved.

Dynamic Programming and Recursion:

Dynamic programming is basically, recursion plus using common sense. What it means is that recursion allows you to express the value of a function in terms of other values of that function. Where the common sense tells you that if you implement your function in a way that the recursive calls are done in advance, and stored for easy access, it will make your program faster. This is what we call Memoization - it is memorizing the results of some specific states, which can then be later accessed to solve other sub-problems.

The intuition behind dynamic programming is that we trade space for time, i.e. to say that instead of calculating all the states taking a lot of time but no space, we take up space to store the results of all the sub-problems to save time later.

Let's try to understand this by taking an example of Fibonacci numbers.

Fibonacci (n) = 1; if n = 0

Fibonacci (n) = 1; if n = 1

Fibonacci (n) = Fibonacci(n-1) + Fibonacci(n-2)

So, the first few numbers in this series will be: 1, 1, 2, 3, 5, 8, 13, 21... and so on!

A code for it using pure recursion:

int fib (int n) {

if (n < 2)

return 1;

return fib(n-1) + fib(n-2);

}

Using Dynamic Programming approach with memoization:

void fib () {

fibresult[0] = 1;

fibresult[1] = 1;

for (int i = 2; i<n; i++)

fibresult[i] = fibresult[i-1] + fibresult[i-2];

}

Are we using a different recurrence relation in the two codes? No. Are we doing anything different in the two codes? Yes.

In the recursive code, a lot of values are being recalculated multiple times. We could do good with calculating each unique quantity only once. Take a look at the image to understand that how certain values were being recalculated in the recursive way:

enter image description here