upGrad

Module Name - Dynamic Programming

Topic Name: Introduction to DP: Coin Exchange Problem

Time Allocation Summary

Element	Slide numbers	Maximum time(min)
Spot Test	4	20
Today’s Agenda	5	10
Introduction to DP	6 - 10	40
Coin Exchange Problem	11 - 12	40
Total Time		120

Spot Test

Let’s take a quick revision assessment of previously taught topics :-

- Algorithm analysis
- Time complexity
- Asymptotic notation
- Time complexities of some very basic algorithms

Today’s Agenda

- Introduction to Dynamic Programming
- Coin exchange problem

Introduction to Dynamic Programming

upGrad

- In the previous modules, you learnt about algorithmic analysis and the divide and conquer technique.
- In simple terms, algorithms are methodological approaches to accomplishing tasks or solving problems.

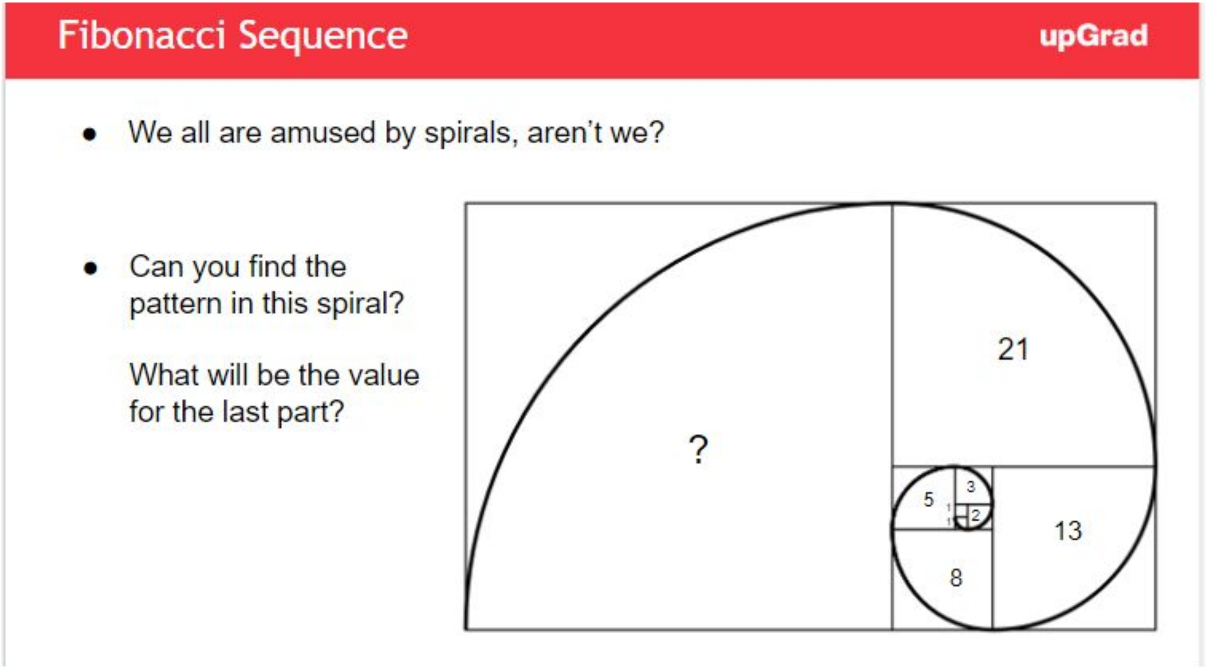
Introduction to Dynamic Programming

upGrad

- In this module, you will learn about dynamic programming, which is a popular technique to solve a particular kind of problem where you are required to find the best possible solution from a number of different solutions.

Introduction to Dynamic Programming

Remember the Fibonacci Sequence we discussed in one of the previous modules. There were multiple algorithms to obtain the sequence. Let's revisit them once again:



Fibonacci Sequence

- We all are amused by spirals, aren't we?
- Can you find the pattern in this spiral?

  What will be the value for the last part?

?
21
5
3
1
1
2
13
8

Introduction to Dynamic Programming

upGrad

The most efficient algorithm we found was:

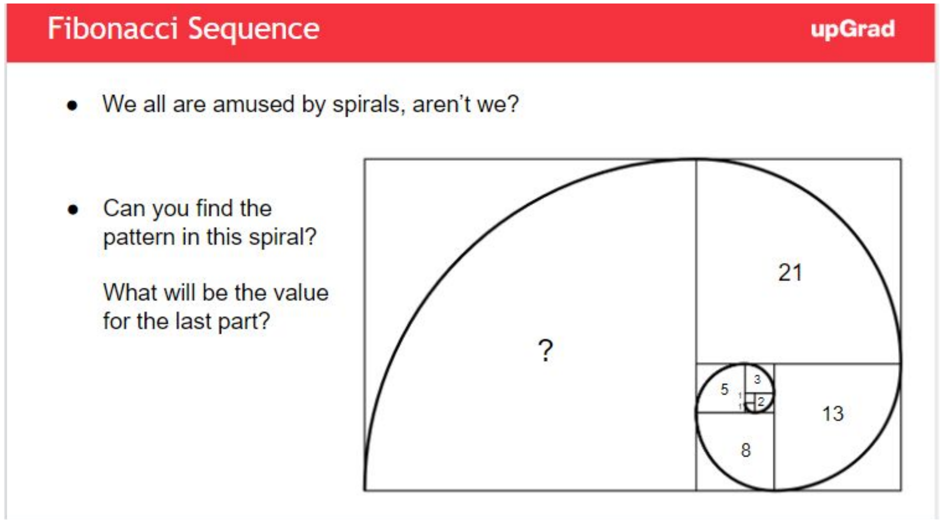
```
public int fibonacci(int n) {
    int a = 0, b = 1, c = n;
    for (int i=2; i<=n; i++){
        c = (a + b)%10;
        a = b;
        b = c;}
    return c;
    }
```

Introduction to Dynamic Programming

upGrad

- This best, linear time solution is a direct application of Dynamic Problem.

- The final solution is obtained by remembering and using the results of smaller similar sub-problems of the same problem.



Fibonacci Sequence

upGrad

- We all are amused by spirals, aren't we?

- Can you find the pattern in this spiral?

  What will be the value for the last part?

?
21
13
8
5
3
2
1
1

Coin Exchange Problem

Let’s start with one of the basic examples of dynamic programming, **The Coin Exchange Problem**.

Let’s say you have coins of different denominations, and you have to pay an amount. What is the minimum number of coins you can use to make this payment?

For example if you need 17 rupees but only have coins of rupee 1, 5, 10 and 20. Which coins will you choose?

Coin Exchange Problem

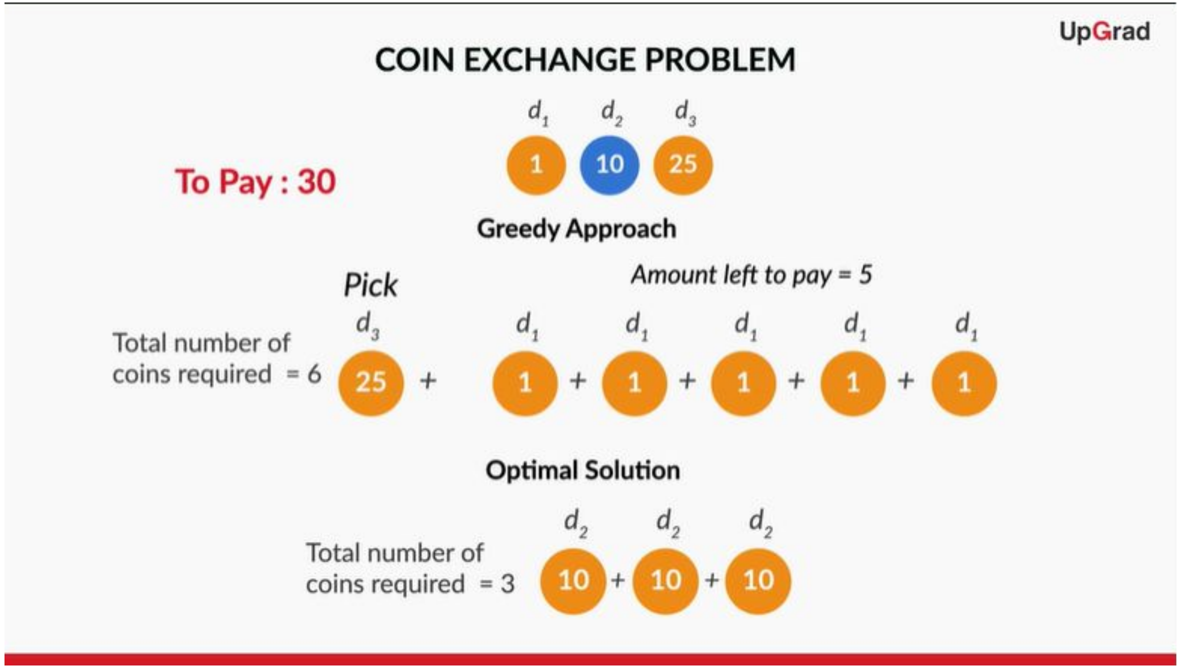
upGrad

Can you think of an algorithm to tackle this problem?

- The most basic solution is to check usability of the coins in the decreasing order of their denominations.
- So, if the highest value coin can be used, then use as much of that denomination as possible and then move on to the next largest denomination.
- If any particular denomination can’t be used, simply check for the next largest denomination and so on.
- **This approach is called the greedy algorithm approach.**

Coin Exchange Problem

We saw the greedy algorithm but it doesn't provide the optimal solution all the time. Let's see the case where we need 30 rupees from the denominations of 1, 10 and 25:



upGrad
#LifeKoKaroLift

Thank You!

Happy learning!