

How to solve greedy problems

Apparently greedy problems are easiest to solve compared to other paradigms like Dynamic Programming. The general rule is to pick the alternative at any point of time which is most beneficial to you.

Let's take this problem. You have infinite number of coins of value 5, 2 and 1. You'll have to make a total sum of 13. Now one has to calculate minimum number of coins needed to get the changes.

We can solve this greedily by taking the highest value coin first and going on similarly.

So we take 5 first now the total remains 8

So we take another 5 now the total remains 3

Now we take a 2 and now the total remains 1

Now we take a 1 to make remaining total 0.

This being said I'll have to say that greedy approach doesn't give the most optimal solution. In some cases it has been proved to always give an optimal solution but not always. Like Dijkstra's algorithm is also takes greedy approach and has been proven to always give optimal solution as long as none of the edges has negative values.

Coming back to the coin change problem mentioned above, greedy will give optimal solution to the denominations mentioned above but if we change the denomination to 5, 3 and 1, it won't give the optimal solution.

Greedy Approach to get change of 13 using 5, 4 and 1 would be $\rightarrow 5, 5, 1, 1, 1$. That is 5 coins. Now if we take Dynamic Programming approach the solution will be $\rightarrow 4, 4, 4, 1$. That will be 4 coins and hence more optimal solution.

Due to this nature of greedy problem to give near optimal solution in lesser amount of time, it has been used to approximate lots of NP hard problems.

