Greedy

As the name implies, a greedy algorithm always chooses the option that appears to be the best at the time. This means it will make a locally ideal choice in the hopes of arriving at a globally optimal answer.

While coming up with greedy algorithm is very easy. To proof that it works you have to prove that problem satisfices these two properties.

For greedy to work

It should have to following properties

- **Greedy Choice Property :** We should be able to find one-first- step towards the optimum solution.
- **Optimum Substructure Property :** Once the first step is taken, We should be able to restructure the rest of the problem as a smaller version of same problem such that induction follows.

Basic Algorithm

1. Take the most optimum step towards the solution
2. Restructure the the smaller problem in the form of original problem
3. Repeat steps 1 and 2 until problem is solved

Time Complexity analysis

Finding the time complexity of any greedy algorithm is very easy.

Time Complexity = O(Finding and Executing Optimum step) * O(Restructuring) * O(No. of restructuring required)