Greedy Algorithms

November 13, 2020

Introduction

- An algorithm design method
- Used to solve combinatorial optimization problems that goes through a sequence of steps
- At each step, the algorithm makes the choice that looks best at the moment (*locally optimal choices*)
- *An example*:

  Make change of 36 Rs, with minimum number of coins.
  Available denominations are [50, 20, 10, 5, 2, 1]

  Greedy Method: To minimise the number of coins, keep on pulling out maximum denomination possible at each step. In this case, First 20, then 10, then 5 and finally 1. Thus at each step we use the optimal choice at that point.

Optimality of greedy algorithm

- Greedy algorithms do not always yield optimal solution. *For example*, assume that the denominations are [10, 5, 4, 1]. To make change of 8Rs, a greedy approach gives 4 coins (5, 1, 1, 1) whereas the optimal is 2 coins (4, 4)
- The problem must have two properties for a greedy algorithm to work:
  1. Optimal substructure
     The optimal solution for a problem contains optimal solutions to the sub-problems
  2. Greedy choice
     By making locally best choices it is possible to construct the globally optimum solution

Optimal substructure

The optimal solution for a problem contains optimal solutions to the sub-problems

For example, Consider the coin changing problem with denominations $[50, 20, 10, 5, 2, 1]$.

For the given value 36, the optimal solution consists of the 4 coins (20, 10, 5, 1).

If we remove a coin from the optimum solution say 20, then the remaining coins (10, 5, 1) forms the optimum solution for the problem $36 - 20 = 16$

Coin changing problem exhibits optimal substructure property.

Greedy choice

By making locally best choices it is possible to construct the globally optimum solution.

Consider the coin changing problem again with denominations [50, 20, 10. 5, 2, 1].

For the optimum solution (20, 10, 5, 1), even if we select the coins in a different order, we will still yield the optimum solution.
Assume the contrary that the coins were selected in a different order and one of the coin say 10, is not in the solution. But in that case, the new solution contains more than one lower denomination coins whose sum is equal to 10. This is not optimal because we can replace the lower denomination coins with a single coin (10) and thus yield a better solution.