Greedy Algorithms

Joyen Benitto

Greedy Algorithms

In this doucment I encompass an introduction to what I believe is one of the most fundamental way of problem solving. "Greed is good. Greed is right. Greed works. Greed cuts through, clarifies, and captures the essence of the evolutionary spirit".

An algorithm is greedy if it builds up a solution in small steps, choosing a decision at every step to myopically to optimize some underlying criterion. The same problem can have multiple greedy solutions based on what measure you are optimizing locally to reach the optimal solution.

When a greedy algorithm could reach a optimal solution or close to optimal it also says that problem could be solved by local decision rules. With the above definition greedy might look like an algorithm that you can use in any case and yes they could be done easily but finding cases in which they work well and and proving that they work well is challenging.

There are primarily two methods to prove that greedy algorithm stays ahead:

1. Establishing that the *greedy algorithm stays ahead*; by means that if we measure the progress of a greedy algorithm in a step by step fasion one sees that it stays ahead of any other algorithm.
2. The *exchange argument* method is more general: one considers any solution and uses greedy approach to adn gradually transform the solution to optimal solution without hurting the quality.

The Interval Scheduling problem

In this subsection we will discuss about the *interval scheduling* problem and how we can optimize it for local solutions.

Consider the set of requests $\{1, 2, 3, 4..n\}$; the i^th request corresponds to an interval of time starting at $s(i)$ and f_i rather than $f(i)$