Greedy Algorithms

When Acting Locally is Acting Globally

Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Fall, 2022

Introduction

Sometimes doing what seems best locally also happens to be what's best globally. Algorithms that follow this pattern are called *greedy algorithms*. They are wonderfully easy to solve, but you can get into a lot of trouble if you think a greedy algorithm will solve a problem when it turns out a dynamic programming approach is best.

Objectives

- Explain the properties of a greedy algorithm.
- Give some examples of greedy algorithms.

Properties of Greedy Algorithms

- They have *optimal substructure* — subproblems have optimal solutions that can be combined to get the main solution.
- They have the *Greedy Property* — We will never regret making a greedy choice locally.

Classic Examples: Coin Change

- Given coins of values 25, 10, 5, 1: make 57 with as few coins as possible.
- Greedy for this version! $57 = 25 \times 2 + 5 + 1 \times 2$.
- Can you break this property?

Classic Examples: Coin Change

- Given coins of values 25, 10, 5, 1: make 57 with as few coins as possible.
- Greedy for this version! $57 = 25 \times 2 + 5 + 1 \times 2$.
- Can you break this property?
- A 20 cent coin will break the greedy property!
- 40 cents $= 20 \times 2$ is optimal, not $25 + 10 + 5$.

In Contests

- Use it if you can, but *be sure*. Otherwise, use Complete Search or DP.
  - Problem statements may give you example data to mislead you into thinking a greedy algorithm will work.
- Learn a few classic algorithms: coin change, load balancing, interval covering
- Preprocessing input can help... e.g., sorting your input first.

Graph Greedy Algorithms

We have already talked about a few graph algorithms that turn out to be greedy.

- Kruscal's MST
- Prim's MST
- Dijkstra's Shortest Path

There are many others.

- Graph Coloring
  - Brook's Theorem: A graph with a vertex of max degree x is colorable in x colors (or $x + 1$ if there is an odd cycle.)
  - Technique: for each vertex v, color v with the smallest available color.
  - What do you thing? Optimal? Not Optimal?

Interval Covering

- You have to cover an interval as best you can with the least number of given sub-intervals.
  - Technique: sort sub-intervals by starting point.
  - Each round, take the least starting point with the max end-point.
  - Repeat until no intervals remain.

Others

- Huffman Coding
- Maximal Product of Elements of an Array
  - $\{-1, -2, 5, 10\} = 100$
  - $\{-2, 5, 10\} = 50$
  - Can you decide the rules?

Others

- Huffman Coding
- Maximal Product of Elements of an Array
  - $\{-1, -2, 5, 10\} = 100$
  - $\{-2, 5, 10\} = 50$
  - Can you decide the rules?
- Take the number of negatives.
  - If even, the product is everything but the zeros.
  - If odd, the product is everything but the smallest magnitude negative and the zeros.
  - There is a special case. What is it?