COMP 3760

Algorithm Analysis and Design

Lesson 16: Greedy Algos

Rob Neilson

rneilson@bcit.ca

Greedy Algos (concept)

Algorithm design technique: Greedy approach

- typically used for optimization problems
- not guaranteed to find an optimal solution, but sometimes it does

General idea:

- given a set of choices/options, always choose the one that is the “best” choice that we can make right now
- the “best” choice is the choice that gets us closest to an optimal solution
- we can consider choices that were made previously
- cannot consider what choices will be made in the future

Example:

- trick or treating:
  - want to maximize the amount of candy you get in 1 hour
  - greedy heuristic: always go to the next closest house
- is the solution guaranteed to be optimal?

Greedy Algorithms (example 1)

- Last lesson we learned how to sort a set of items topologically

- The second algorithm went something like this:
  - `identify a v ∊ V that has indegree = 0`
  - `delete v and all of its edges`
  - `when all vertices have been deleted, the topo sort order is given by the order of deletion`

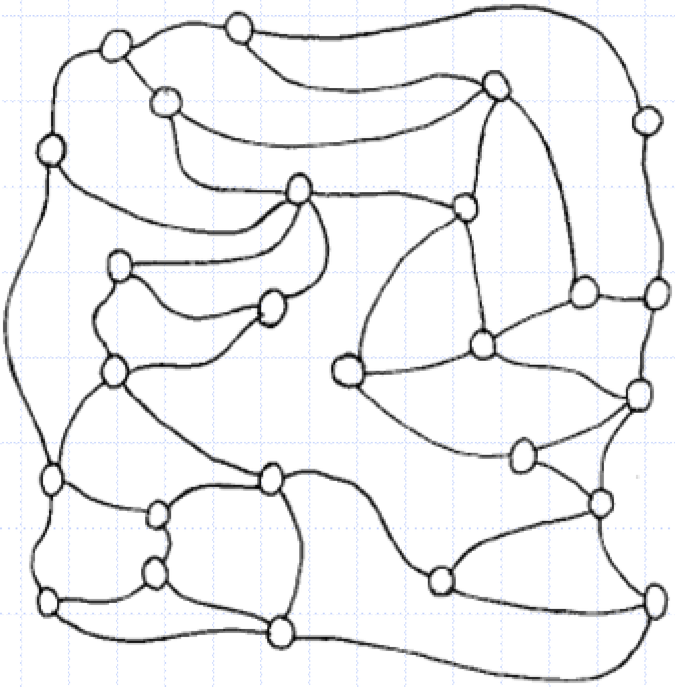
- *This was a greedy algorithm!*

- At each step we were just doing the "best thing" – based on what we knew at the time.

- In this case the greedy algorithm always produces an optimal solution!

Greedy Algorithms (example 2)

- Map coloring problem: color a map with as few colors as possible such that no two adjacent areas are the same color

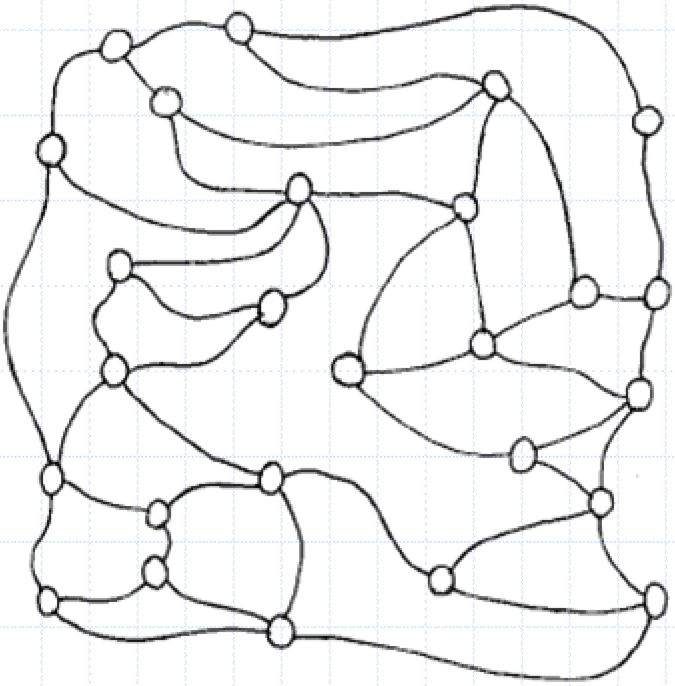


- Greedy solution:
  - choose a color that has not been used yet
  - color as many areas as possible with this color
  - repeat above two steps until colored

Will this always result in an optimal solution?

Greedy Algorithms (examples)

- Coffee shops:
  - assume vertices are cities, edges are roads
  - want to know minimum number of coffee shops (and where to place them) such that every city is at most one hop away from a city with coffee
  - note: if we generalize this problem it is known as a “dominating set problem”



- Greedy solution:
  - find city with most neighboring cities
    - put a coffee shop there
    - remove the city and its neighbors from the graph
  - repeat until all cities are covered

The End