MSSC 6000
Feb 2, 2022

Announcements:
→ HW 1 due next Wed on D2L

Lecture 3 - Greedy Algorithms

A lot of topics in this class are in the category of "problem solving paradigms".

A catalogue of ways to approach new problems. "heuristics"

What is a problem?
You have input data and/or constraints.
One question might be:
- Is it possible to satisfy all of the constraints?

Ex: Every year, the NFL has to come up with a season schedule.

There are many constraints!

- 32 teams in 2 conferences
- Each conference is split into 4 divisions of 4 teams each.

Each team plays:

- all 3 division rivals, twice
  one home / one away
- each team in some other division in the same conference
  two home / two away
- four teams in the other conference
  two home / two away
- two more teams in their own conference 1H / 1A

- stadium constraints
- TV constraints
- holiday games
- bye week

Q: Can this be done?

A: Yes. The NFL says they use 100s of computers. They come up with ~1000 good schedules. Humans pick the best out of those.

- Another type of question:
  Which solution is optimal?
  (lowest cost, highest value, etc)

Ex: If Amazon has 100 packages to deliver to different houses in Milwaukee, and 5 delivery vans, which route
- uses the least gas?
- travels the fewest miles?
- takes the least time?

or some combination of these

Greedy Algorithms

Vague Definition: A greedy algorithm is a way of solving a problem that builds up a solution bit by bit, always picks the next bit that is the best, even if that leads to a suboptimal solution in the end.

They are:
- normally lightning fast
- much better than random solutions

- sometimes pretty bad,
  sometimes pretty good,
  sometimes provably optimal,
  depending on the problem.

<u>Ex:</u> Giving change - How does a cashier give change? Suppose you owe \$3.27 and pay with \$20. They start giving you bills and coins from largest to smallest.

$\$20 - \$3.27 = \$16.73$

~~\$100~~	~~\$50~~	~~\$20~~	\$10	\$5	\$1	\$0.25	\$0.10	~~\$0.05~~	\$0.01
			1	1	1	2	2		3
			6.73	1.73	0.73	0.23	0.03		0

→ 1 \$10, 1 \$5, 1 \$1, 2Qs, 2Ds, 3Ps

= 10 things. This is a greedy algorithm.
Is it optimal?
fewest # of bills/coins

<u>Theorem:</u> For the US currency denoms listed above, the cashier's algorithm is optimal.

Proof: To simplify things, let's assume denoms of 1, 5, 10, 25 cents. Suppose we are making x cents.

Lemma: The optimal solution will have ≤ 4 pennies.

Proof: If the sol. had ≥ 5 Ps we could replace 5P with 1 N and get a better solution.

Lemma: The optimal solution will have ≤ 1 N.

(same proof)

Lemma: The optimal sol. will have #N + #D ≤ 2

Proof: 2N: bad

1N + 2D: bad → 1 Q

0N + 3D: bad → 1N + 1Q

Main proof, by induction:

Base case: 0¢ sol: 0 coins optimal ✓

Now assume we're making x ¢ worth of change.

Case 1: $x < 5$, x pennies, optimal ✓

Case 2: $5 \le x < 10$, must have 1N
(because the lemma says no more than 4 pennies)
1 N + (greedy sol. for $x-5$ cents) ✓
optimal by induction

Case 3: $10 \le x < 25$, similar to Case 2

Case 4: $x \ge 25$: must have 1Q
(or else >4 P or >1 N or 1N+2D, or 3D)
sol: 1Q + (greedy sol. for $x-25$ cents). ∎

Is the cashier's algo. optimal for *any* set of denominations? No.

Ex: US Postage Denominations
1, 2, 3, 5, 10, 20, 35, 36, 55, 65, 75, 95, 100, 120, 200, 500, 795, 1000, 2635.

To make 72¢: 65 + 5 + 2 3 stamps
Better: 36+36 = 72 2 stamps

Greedy $\neq$ Optimal

Throughout the course we're going to learn about a catalogue of problems that model all kinds of real world problems that you might face.

Problem #1: Interval Scheduling