MSSC 6000

Feb 14, 2022 – Day 10

Lecture 3 - Greedy Algorithms (continued)

* There is no known greedy algo that is optimal.

How long would brute force take?

If there are n requests, then the # of subsets of requests is 2^n. To check them all takes exponential time.

"big-O notation" – roughly how many steps an algorithm has to do

WIS is $O(n \cdot 2^n)$

Later lecture on "dynamic programming", it can be done in $O(n \cdot \log(n))$

great!

Problem #4 – Knapsack Problem

You have n items that each have some value v_i and some weight w_i. You have a knapsack that can carry a total weight of C. What combination of items has the highest total value while having the sum of the weights $\leq C$.

Ex: Capacity = 10

	item	weight	value	
	1	8	13	1.625
✓	2	3	7	2.33
✓	3	5	10	2
	4	5	10	2
✓	5	2	1	0.5
	6	2	1	0.5
	7	2	1	0.5

Some possibilities:

* Items 1, 5
  weight = 8 + 2 = 10 ✓
  value = 14

* Items 2, 4, 7
  weight = 10 ✓
  value = 18

* Items 3, 4
  weights = 5 + 5 = 0 ✓
  values = 10 + 10 = 20
  optimal for this case

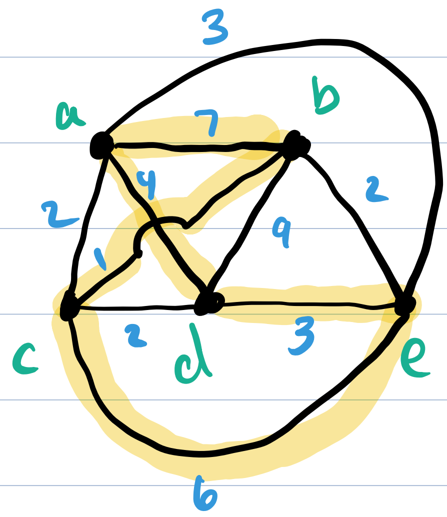
* Greedy possibilities:
  - best = lightest item (value 10 in our example)
  - best = highest value (value = 14)
  - best = most value dense: $\frac{\text{value}}{\text{weight}}$ (value = 18)

None of these are the optimal value of 20.

We'll learn that dynamic programming can solve it in polynomial time.

Problem #5 - Traveling Salesman Problem (TSP)
There are n cities that a salesman needs to visit, and then return home. What is the shortest route that visits each city exactly once and then returns back to the starting place?

More formal: Consider a weighted graph G. Which ordering of the vertices gives you the smallest sum of the edge weights?



$a \to d \to e \to c \to b \to a$

$4+3+6+1+7 = 21$

$e \to c \to b \to a \to d \to e$

Same 5 edges $=21$

$a \to c \to b \to e \to d \to a$

$2+1+2+3+2$

$=10$