Greedy Algorithms

Greedy is the antithesis of Brute Force.
↳ DON'T LOOK @ ALL OPTIONS, MAKE "LOCAL" GREEDY CHOICE THAN "SEEM" TO BE RIGHT.

2 Greedy Algs we've seen so far are: KRUSKALS, PRIM.

DEFINITION OF AN ALGORITM:
FINITE SET OF STEPS THAT CORRECTLY SOLVES A PROBLEM.

BIGGEST DANGER IN DESIGNING THESE - NOT SOLVING ALL INSTANCES OF THE PROBLEM CORRECTLY!!!

HARDEST PART - PROVING CORRECTNESS

MOST PROOFS - "EXCHANGE ARGUMENT"
= "PROOF CONTRADICTION"

MAKING CHANGE

Example: 1, 5, 10, 50, 100, 1000

General: $d_1, d_2, d_3, \ldots d_k$

where $d_i \mid d_{i+1}$ for all i

$1 \leq i < k$.

Given a set of denominations and a value to make change for, determine the fewest # of coins necessary:

$n = 968$

968/1000 → 0 of 1000 coins

968/100 → 9 of 100 coins

968%100 → 68/50 → 1 of 50 coins

68%50 → 18/10 → 1 of 10 coins

18%10 → 8/5 → 1 of 5 coins

8%5 → 3/1 → 3 of 1 coins

(0, 9, 1, 1, 1, 3) → my ans

(0, 9, 0, _, _, _) → alt ans

first spot w/ mismatch

→ ONLY WAY YOU CAN GET 50¢ IS BY combining multiple ones of these.

If not divis

1, 8, 10

16¢

COUNTER EX

SINGLE ROOM SCHEDULING

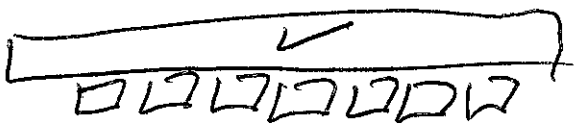
Scheduling a single room

Input - Bunch of requests

[50, 150)	[100, 160)
[10, 30)	[151, 180)
[5, 180)	[150, 157)
[45, 57)	[160, 180)
[60, 90)	

Output - Maximum # of satisfiable request.

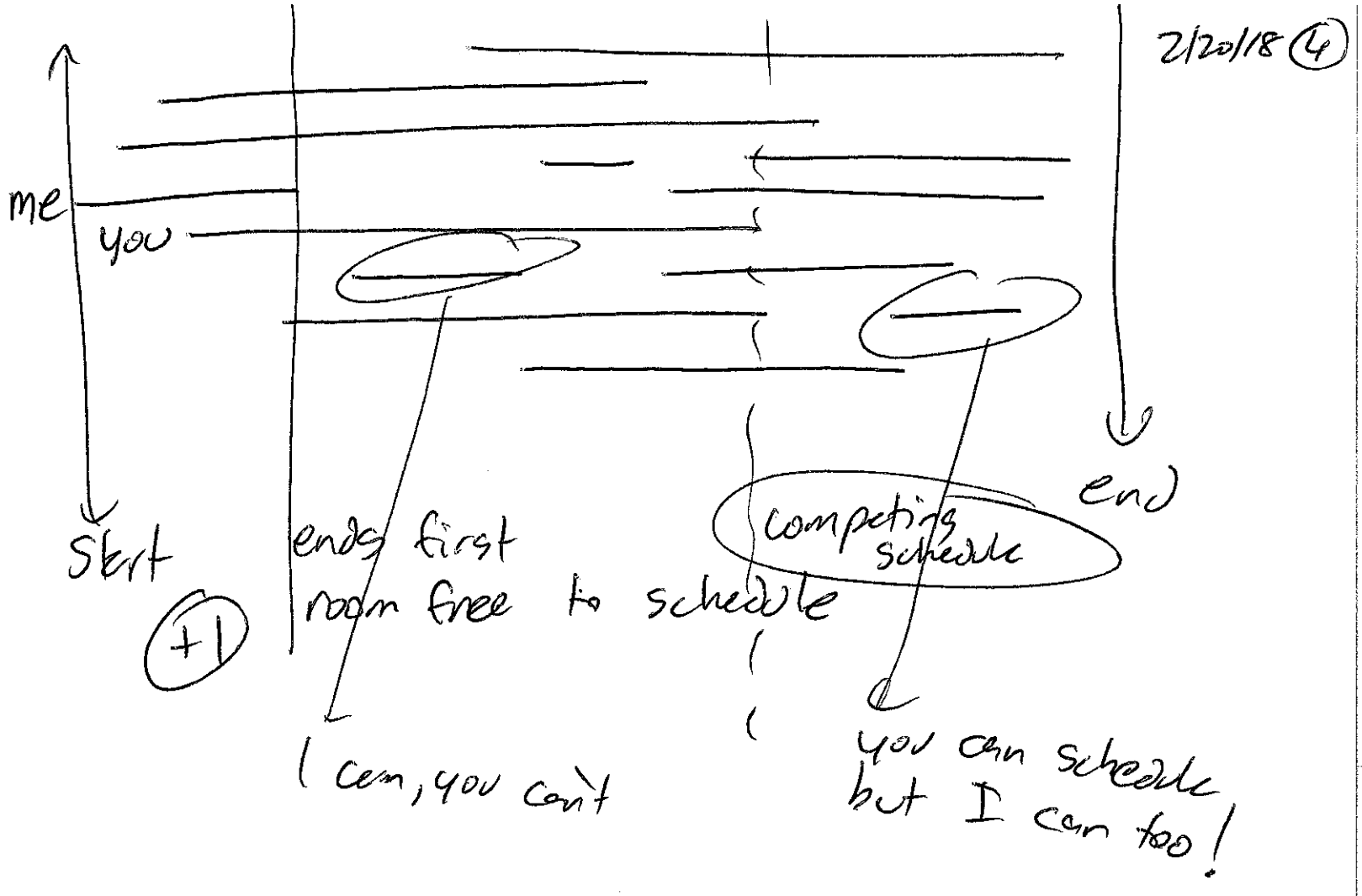
Brute force run-time ~ $O(2^n)$
(look @ each subset)



3 POSSIBLE ALGS

1. Sort requests by event length.
2. Sort requests by start time
3. Sort requests by end time.

Consider each event, one by one, add it to the schedule if it doesn't conflict with anything else in the schedule.



THINK ABOUT –

maximize ~~# of minutes room is used~~.

Doesn't have a greedy solution!!!

GOOD TO BRAINSTORM
TEST CASES TO BREAK
POTENTIAL IDEAS!

MULTIPLE Room SCHEDULING

Input - list of events (start, end)

Output - min # of rooms to schedule ALL events.

ALG

Sort events by start time
When you get to an event put it in the ~~ea~~ lowest # empty room

A - 5, 180
B - 10, 30
C - 45, 57
D - 50, 150
E - 60, 90
F - 100, 160
G - 150, 157
H - 157, 180
I - 160, 180

Room 1: A (160)
Room 2: B, C, E, F (160 ~~90~~ ~~57~~)
Room 3: D, G, I (157 ~~150~~)
Room 4: H (180)

Imagine $n = 10^5$

Sort is $O(n \lg n)$

Have to find earliest room that is free

Priority Queue
Obj rooms (ID, time free)

$O(\lg n)$ for finding 1 min
total → $O(n \lg n)$
(for each → find min; find min)

2/20/18 (6)

Sweep Line Idea

A - 5, 180
B - 10, 30
C - 45, 57
D - 50, 150 ✓
E - 60, 90
F - ✓100, 160
✓G - 150, 157
H, ✓151, 180
I - 160, 180

Create 2 objects for each event
① (start, 1)
② (end, -1)
Sort the 2n objects by time (1st value)
Break ties by having -1 go first.

(5, +1)	(60, +1)	(157, -1)
(10, +1)	(90, -1)	(160, -1) → (160, 1)
(30, -1)	(100, +1)	(180, -1)
(45, +1)	(150, -1)	(180, -1)
(50, +1)	(150, +1)	(180, -1)
(57, -1)	(151, +1)	

CONTAINERS

"Extra Programs"

B
B
C
C

A B C C

Shipments: A B C C C B B C B D B A

1 2 3 4 V V

B
B
B
B

C
C
C

A
A B C D
1 2 3 4

A B C

Place B → A B B
Place B → A B C

Goal: Minimize # of stacks

Input: D C B A → A B C D (1)

① Go through the boxes in order

② Place each box of the "minimum" valid top. If none exists add a new stack.

A C M R V Z

set O →
A C M O V Z
A C M R O Z
A C M R V O