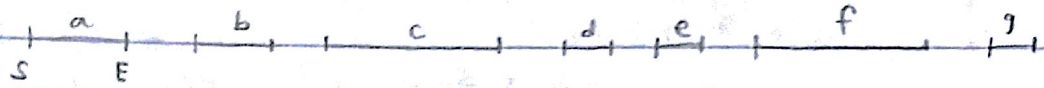
ex: scheduling problem 1/12/16

Given a set of intervals



each interval has a starting + ending point

$S_a \leq F_a$

a set of tasks that overlap -- choose which to do

cannot partially do a task

can't change start + end times

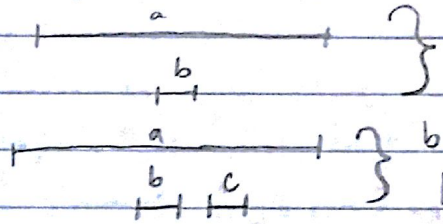
length of interval is fixed

can't pick overlapping intervals

find: maximize number of non-overlapping interval

all equally important

length is unimportant



exhaustive search:

number of subsets: 2^n

must verify if legal (non-overlapping)

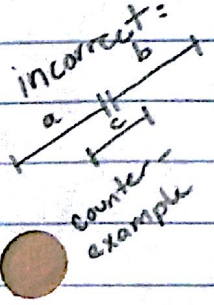
cardinality

time complexity: $O(2^n T_n)$

greedy algorithm:

small intervals probably won't interfere w/ other tasks

only pick small intervals + throw everything else out



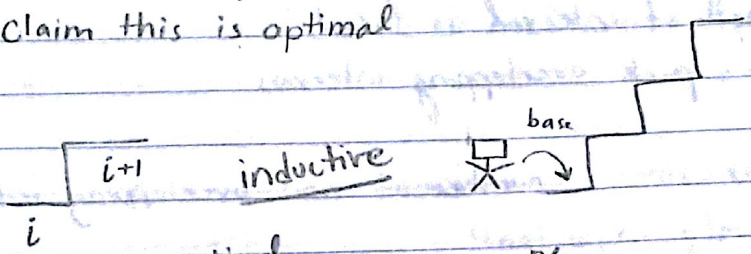
also greedy

Plane Sweep: (use w/ geometries)
left to right
choose first event + throw out everything else

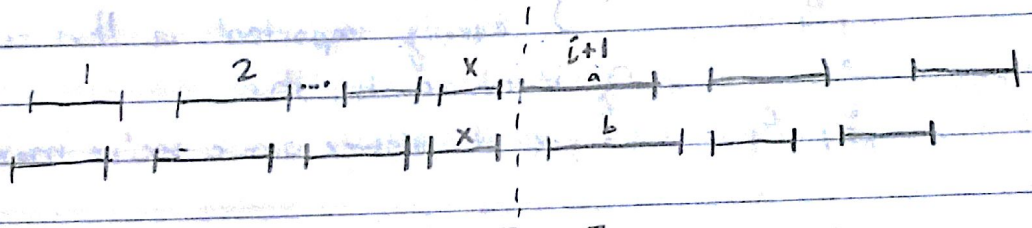
not optimal

idea: pick intervals w/ minimal conflicts

☆ idea: Plane sweep but start w/ ends first
↳ claim this is optimal



number of optimal solutions $\leq 2^{n/2}$



$E_a \leq E_b$

$S_b \geq E_x$

$S_a \leq E_x$

base case: start w/ empty set then next is i+1

in an array sort start and end times together

run time analysis:
take end times then sort them $O(n \log n)$
pick the one that ends first
get rid of any overlapping intervals $O(n)$

time complexity: $O(n \log n)$