

COP3SD3 9/28/26
9

10/1 10/6
① Pick whiteboarding schedule - R, T

② Radix Sort

③ Greedy Algorithms

10/1 R 10:45 Dhruv

10/1 R 11:15 Aidan

10/6 T 10:45 AJ

10/6 T 11:15 Andrew

Radix Sort - Relic

n objects (k digits) b
Multi-digit numbers w/ ~~2~~ symbols
 $0, 1, 2, \dots, b-1$

$d_{k-1} d_{k-2} d_{k-3} \dots d_0$

$O(n)$
for a single
stable sort
total runtime
 $O(nk)$
 $k = \# \text{ digits}$

643	261	512	242
261	242	919	261
896	512	626	344
344	862	242	512
242	643	643	626
512	344	261	643
626	896	862	862
919	626	896	896
862	919		919

Stable sort ↑ Stable sort ↑

Greedy Algorithms

An algorithm that doesn't take into all possibilities, makes a locally "greedy" choice, but still produces the correct answer.

What's easy (usually): If I give you the algorithm, implementing tends to be easier than other strategies

What's hard: ① Proving correctness!

★ ② Figuring out which "greedy" choice is the right one to make.

③ Determining IF there exists a greedy algorithm to solve a problem!

Coin Changing

$d_0, d_1, d_2, d_3, d_4 \dots$

1, 5, 10, 50, 100

$d_i \nmid d_{i+1}$ for all i $d_i \neq d_{i+1}$

Make change for 673¢ minimize # coins

$$x_4 = 6 \times 100$$

$$x_3 = 1 \times 50$$

$$x_2 = 2 \times 10$$

$$x_0 = 3 \times 1$$

$$\underline{12}$$

Compute

y_4

y_3

y_2

y_1

y_0

$$y_3 \neq x_3 \quad y_3 < x_3$$

Competing has to
make up
 $(x_3 - y_3) d_3$
cents
elsewhere

Single Room Scheduling

	$[s_0, e_0]$	③ 4 7	YES
	$[s_1, e_1]$	④ 3 8	X
	$[s_2, e_2]$	⑤ 7 9	YES
		② 1 5	X
		⑧ 1 20	X
		① 3 4	YES
		⑦ 12 18	X

Sort by end
time +
greedily
schedule

multiple room scheduling

Scene input as 1st, everything must be scheduled

- ⑥ 4 7V
- ④ 10 15V
- ④ 3 8V
- ⑥ 7 9V
- ① 1 5V
- ② 1 20V
- ③ 3 4V
- ⑨ 12 18

GOAL: Minimize # Rooms

		[10, 15)		
		[4, 7)	[12, 18)	
	[7, 9)			
	[1, 5)	[1, 20)	[3, 4)	[3, 8)
	R1	R2	R3	R4

Fractional Knapsack

Max Weight W

Items V_i value per unit w_i = amt avail

$W = 20$ lbs

	<u>Value / unit weight</u>	
1	6	3
2	9	7
3	1	30
4	4	6
5	7	2

$$\begin{array}{r}
 63 \\
 + 14 \\
 18 \\
 24 \\
 + 2 \\
 \hline
 121
 \end{array}$$

2 lbs
 $v = 1/1b$

Containers (2007 WF)

A smallest, B 2nd

Z largest

Valid stack

A
D
D
D
M

A, C, M, D, B, C, Q, B, C, A, D

	B	D
<u>A</u>	<u>C</u>	<u>M</u>
S1	S2	S3

old $\rightarrow$ BD
 $\rightarrow$ BC

for each value,
 place on stack
 with min value $\geq$ to it.

Planting Trees + Fylyane Thursday

Did Verbelly!

Last Alphabetic Subsequence

* S, A, P, Q, R, B, M, R, C, P, N, K, L, B, T, A, C, B

Subsequence
PRL
 SAP
 SQR

A
S
 Stack

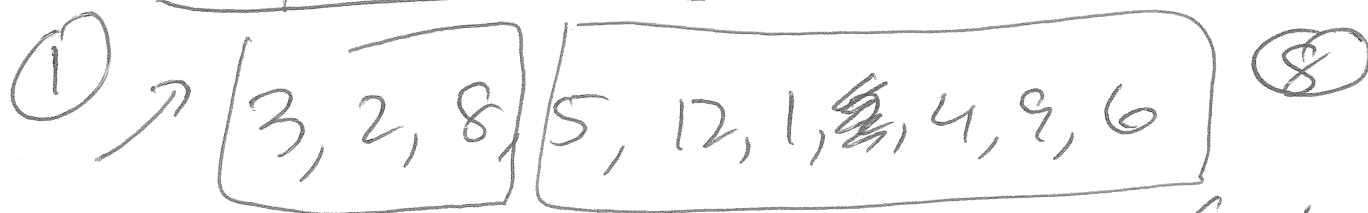
S
P
Q
R
R
R
R

P
R
Q
S

B
A
C
T

Up + Down

3 + 3 + 2



Only allowed
swap of
adj elems

Goal → make
up
down

1, 2, 3, 5, 8, 12, 9, 6, 4

Minimize # swaps

Min # swaps (adj elems) to sort an
array = # of inversions

3 2 1

→ 3 inversion

(3, 2), (3, 1), (2, 1)

~~OBSERVATION~~ OBSERVATION #2: We don't know
where peak will be!

Try each spot / split point

add up inversions

for min ans