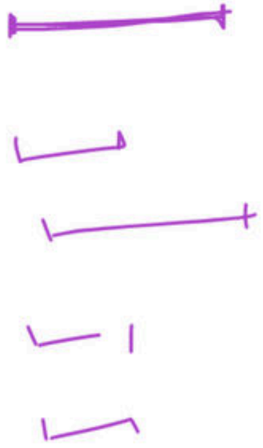
Problem Solving

Course on Game Theory and Greedy Algorithms

Triveni Mahatha • Lesson 3 • May 16, 2021

Greedy solution May not work



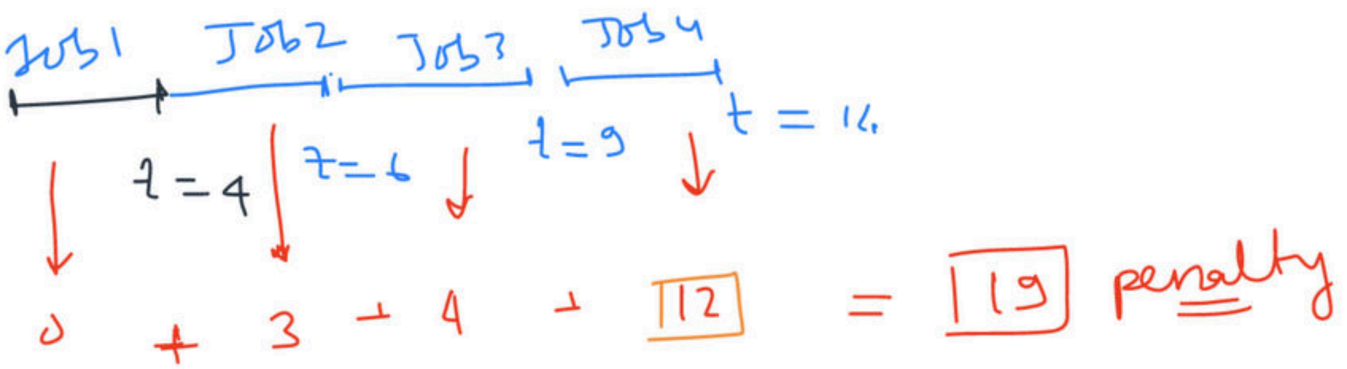
t_i, d_i penalty = $d - d_i$ if $d > d_i$

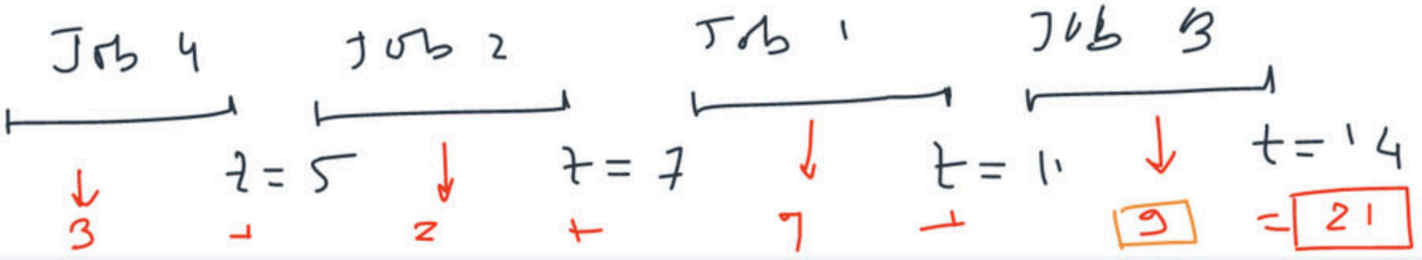
* perform them such that the penalty is min.

The order in which these tasks should be performed??

Day starts at $t = 0$.

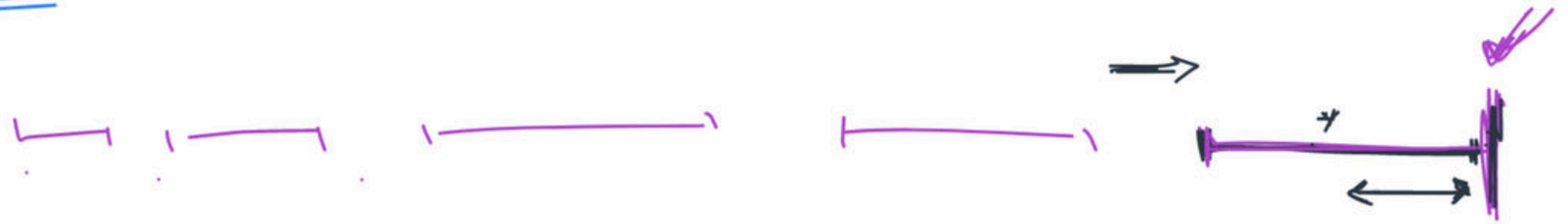
	t_i	d_i
Job 1	4	4
Job 2	2	3
Job 3	3	5
Job 4	5	2



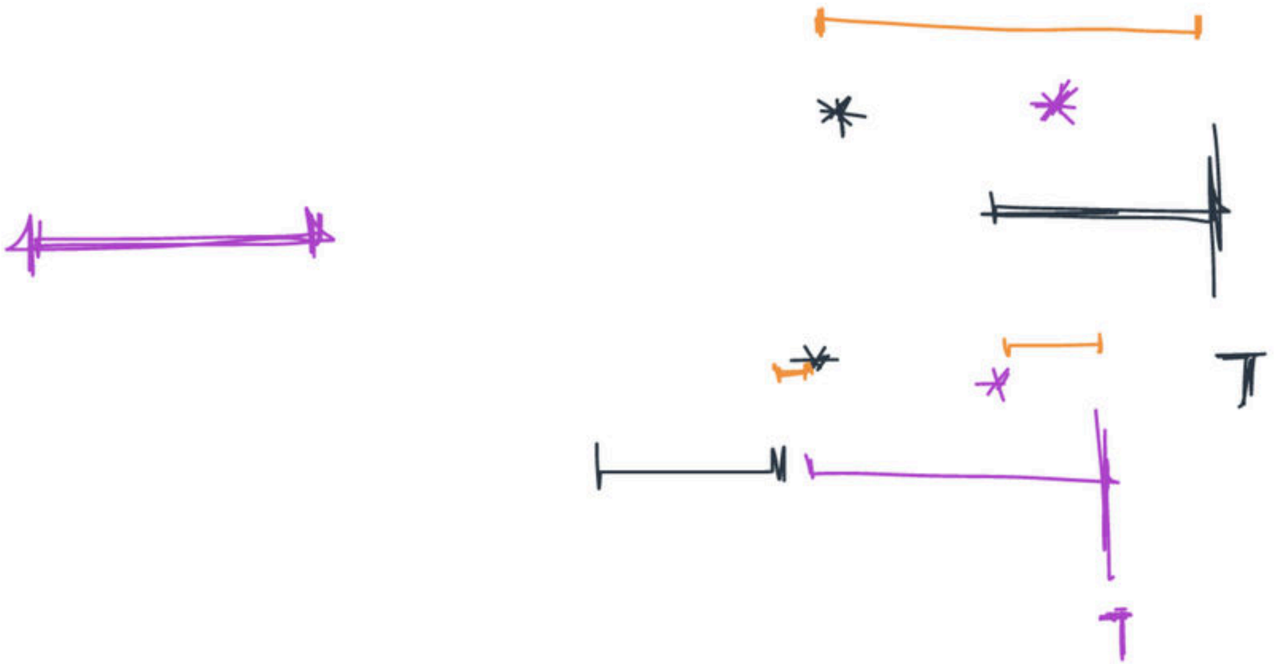


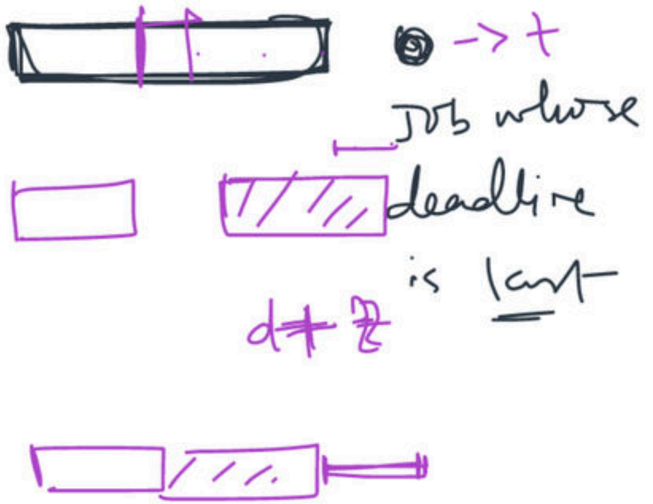
Modification : What if we want to minimize the maximum penalty.

solution!?



$T = \sum t_i$

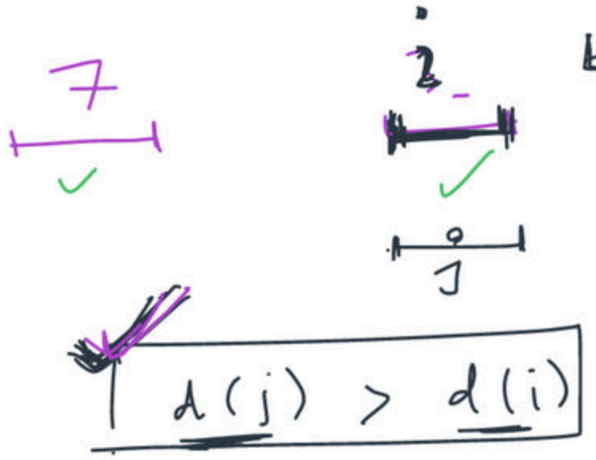




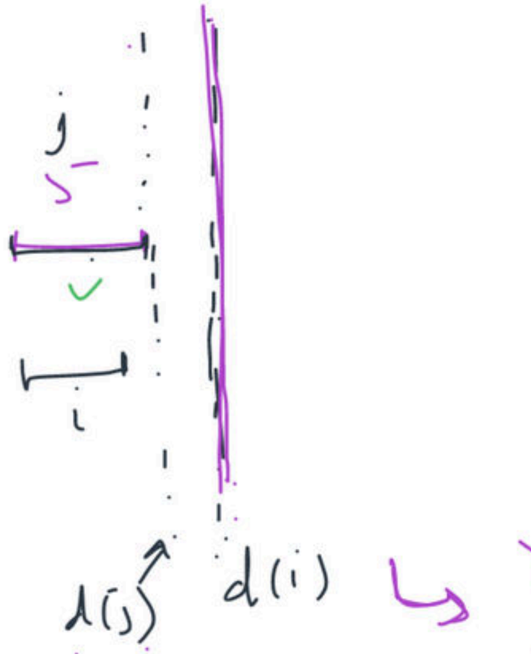
You want to earn maximum possible profit.

NOTE: All of them takes same time to finish. $t_i = 1$

1) Sort in dec order of profit.
2) Now pick a job and do it as late possible but within the timeline.



If $A(j) < d(i)$



↳ You put the guy at the end without losing any profit

'S' ⟶ 'S1'

↑ you removed the jobs which don't earn profit.

Therefore from the exchange argument, it always makes sense to put a job 'i' as late as possible provided that still earns you profit.

modification

How do you schedule to maximize the profit !?

⇓

* This cannot be solved greedily.

⇓

"Dynamic Programming"

Huffman Coding → Lossless compression

"abc" → 3×8 = 24 bits

'A byte long'

'c' → 8 bits

a → 2
b → 3
.
.
. → 12
z → 8

⟹ on an average the total length is minimized.

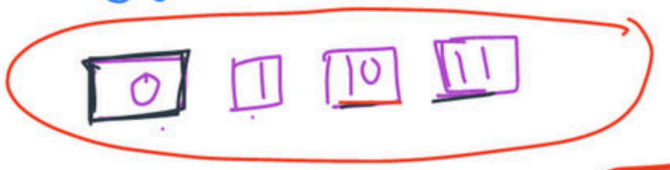
aaaaaaa → 7

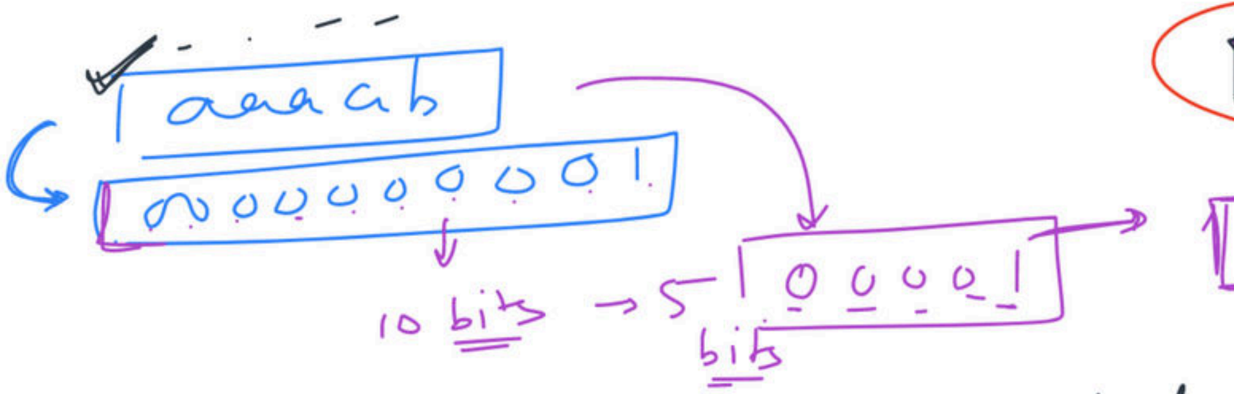
8 + 8 = 16 bits

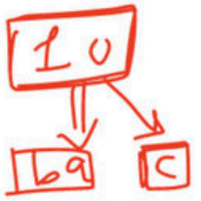
8×7 = 56 bits

$\Sigma = \{a, b, c, d\}$

00 01 10 11







How to assign bit sequences to the alphabets optimally??

To ensure no ambiguities, we need to make sure that bit seq of any character is not a prefix of another bit seq.

$\Sigma = \{ \ldots \ 'x' \ \ldots \ 'y' \ \ldots \ 'z' \}$

$\Rightarrow$ prefix codings

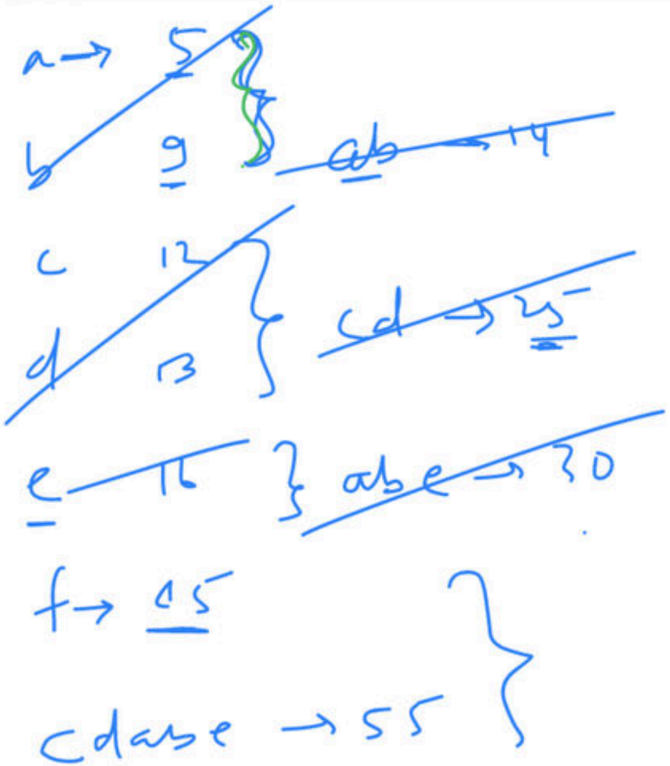
ascii characters are prefix encoding or not!! → Yes

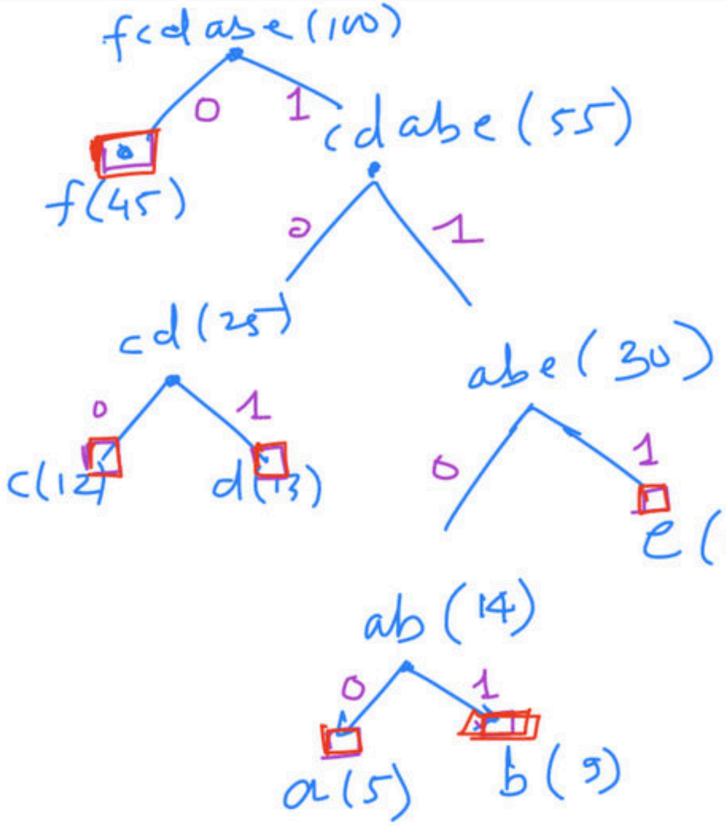
How to ensure optimality??

& bit length

Σ	a	b	c	d	.	.	.
f	f_1	f_2	f_3	f_4	.	.	.

How to determine the bit seq for each of the characters??





Huffman Tree

3x	100	a → 5 × 4	= 20
3x	101	b → 9 × 4	= 36
3x	100	c → 12 × 3	= 36
3x	101	d → 13 × 3	= 39
3x	111	e → 16 × 3	= 48
1x	0	f → 45 × 1	= 45
			224

300 bits for normal bit rep.

$\frac{76}{300} \times 100 = 25.3\%$

You're a robber and broke into a grocery store which contains 'n' kind of grains (rice, wheat etc). Amount of each grain is given and also its monetary value. Your bag has a size 'W'. What is the maximum monetary value can you steal from the store??

	rice	wheat	x	y	z
weight	5	4	5	3	4
price	5	12	13	17	2
	4	3	2	1	5

W = 39

What is the max monetary value that you can steal??

W = 11

3 + 5 + 3 = 11

17 + 13 + $12 \times \frac{3}{4}$ = 37 ✓

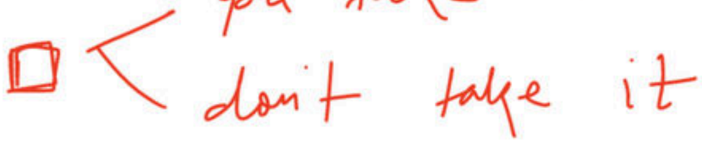
3 y + 4 wheat + 4 unit of 'x' = 17 + 12 + $\frac{13}{5} \times 4$ = 39.4

4 = 4

Ans: Sort according to price/weight (price per unit wt)
then pick them one by one.

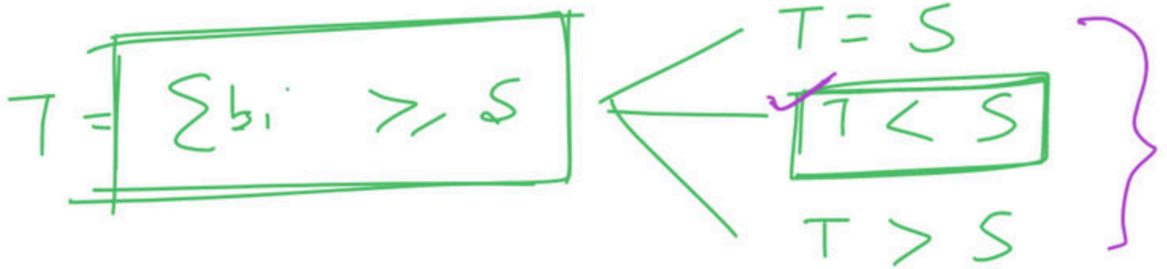
If splitting/breaking is not allowed then what!!

→ Greedy fails || 0-1 knapsack problem → dp



b_i time to destroy that letter

burn as little as possible, but the burning should go on to at least 's' unit of time.



↳ Minimize $\|T-S\|$.

$T_{(.)} = O(n' g_n)$