Reader

	1	2	3
1	A	B	C
2	D	E	F

→

	1	2
1	A	D
2	B	E
3	C	F

	1	2	3
1	A	B	C
2	D	E	None

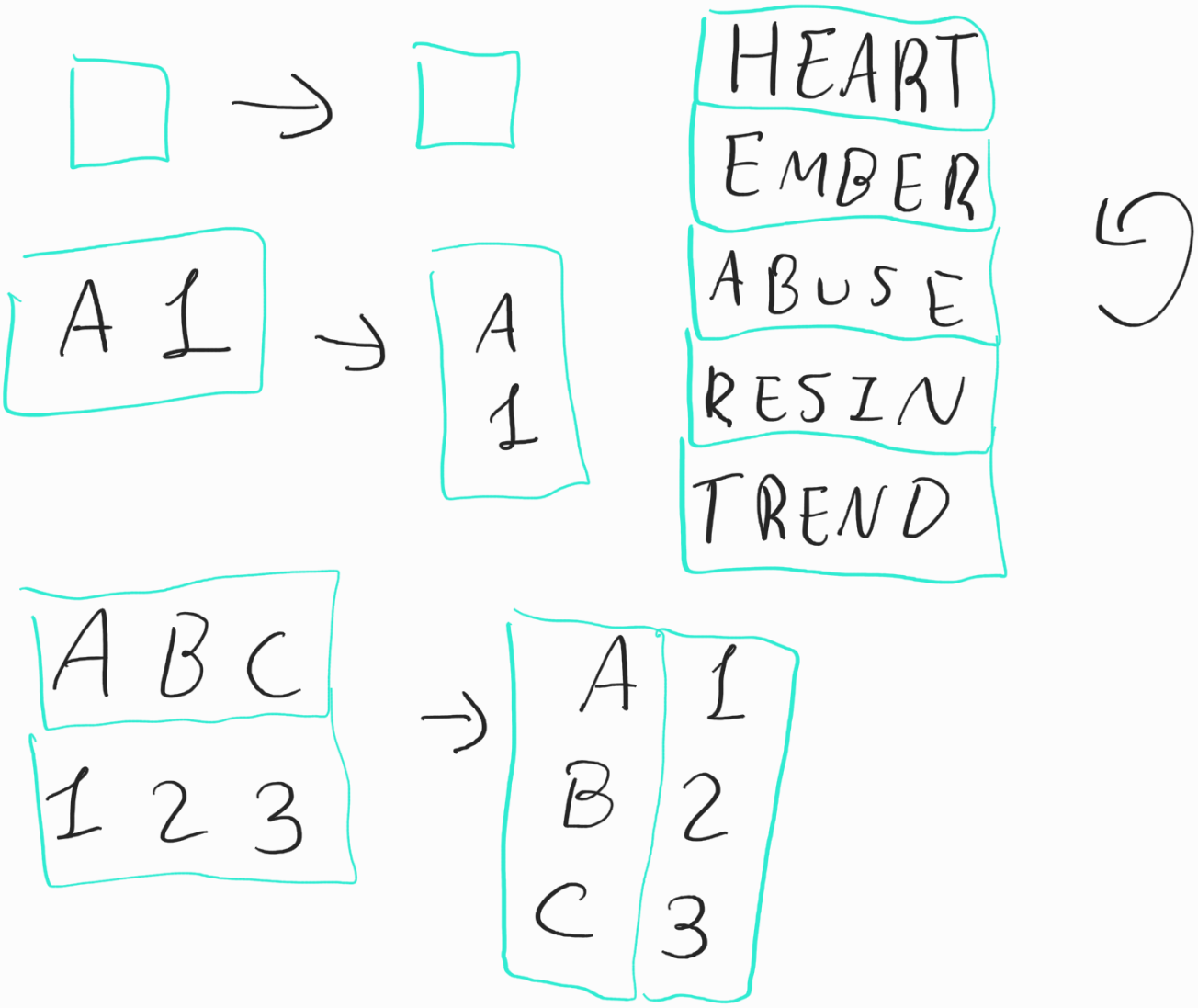
→

1	A	D
2	B	E
3	C	None

1	2	3
A	B	None
D	E	F

→

1	A	D
2	B	E
3	–	F



Program needs to know max row length
Then it needs to fix rows with no
full max row length and iterates
over fixed matrix to get the new one.

A single book costs \$8

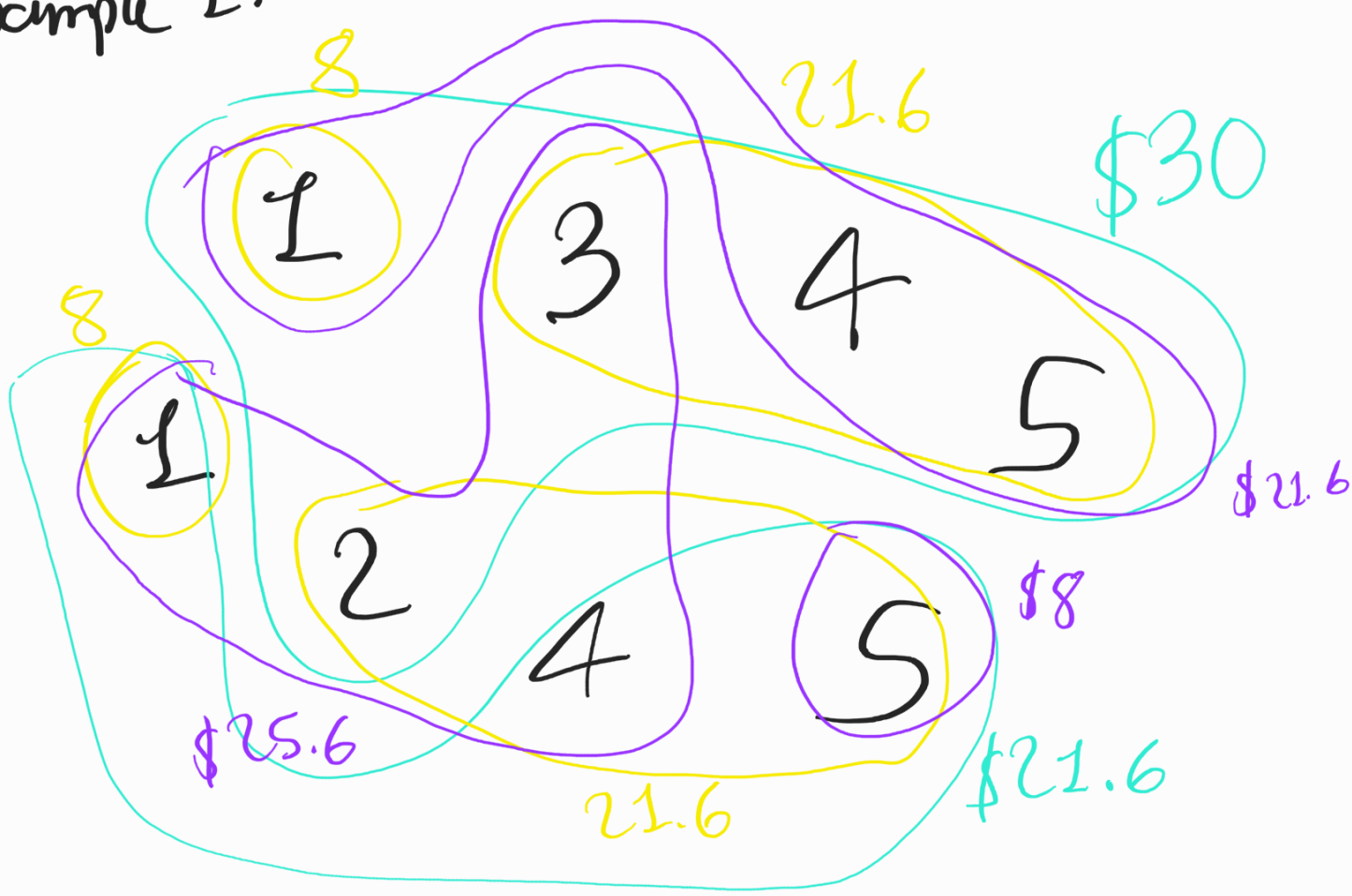
2 diff → 5% = 16 × 0.05 = 0.8 disc

3 diff → 10% = 24 × 0.1 = 2.4 disc

4 diff → 20% = 32 × 0.2 = 6.4 disc

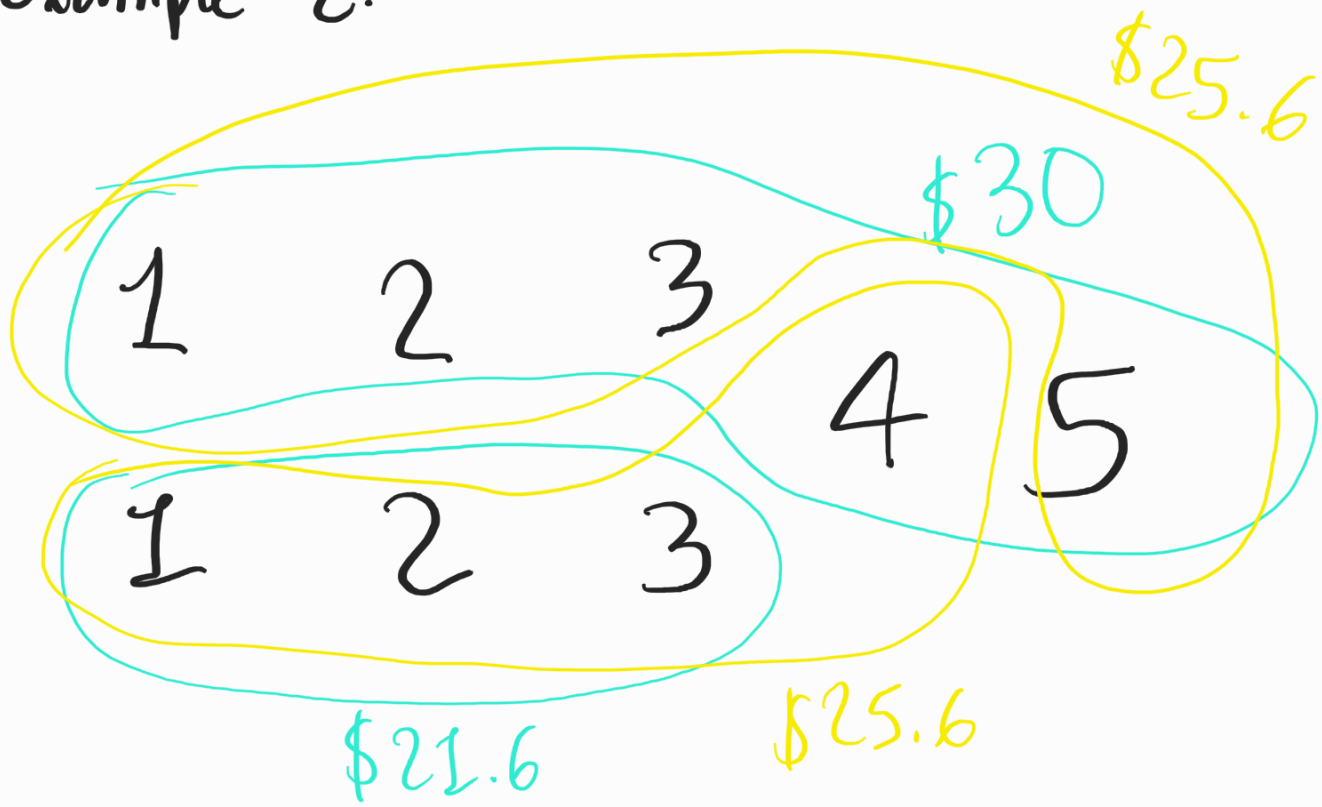
5 diff → 25% = 40 × 0.25 = 10 disc

Example 1:



\$51.6 \$59.2 \$55.2 ...

Example 2:

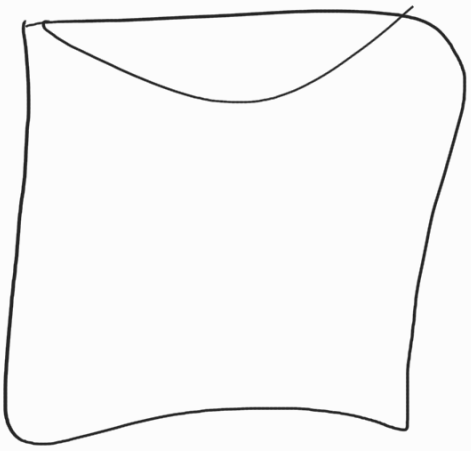


$51.6 $51.2 ...

It could be implemented by using brute force but that is not the idea because of huge calculations and complexity...

Dynamic Programming Aproach:

I think this problem is quite similar as knapsack problem...



$w = 7$

val	1	4	5	7
weight	1	3	4	5

$n = 4$

Val	weight		max weight							
			0	1	2	3	4	5	6	7
0	0		0	0	0	0	0	0	0	0
1	1		0	1	1	1	1	1	1	1
4	3		0	1	1	4	5	5	5	5
5	4		0	1	1	4	5	6	6	9
7	5		0	1	1	4	5	7	8	9