

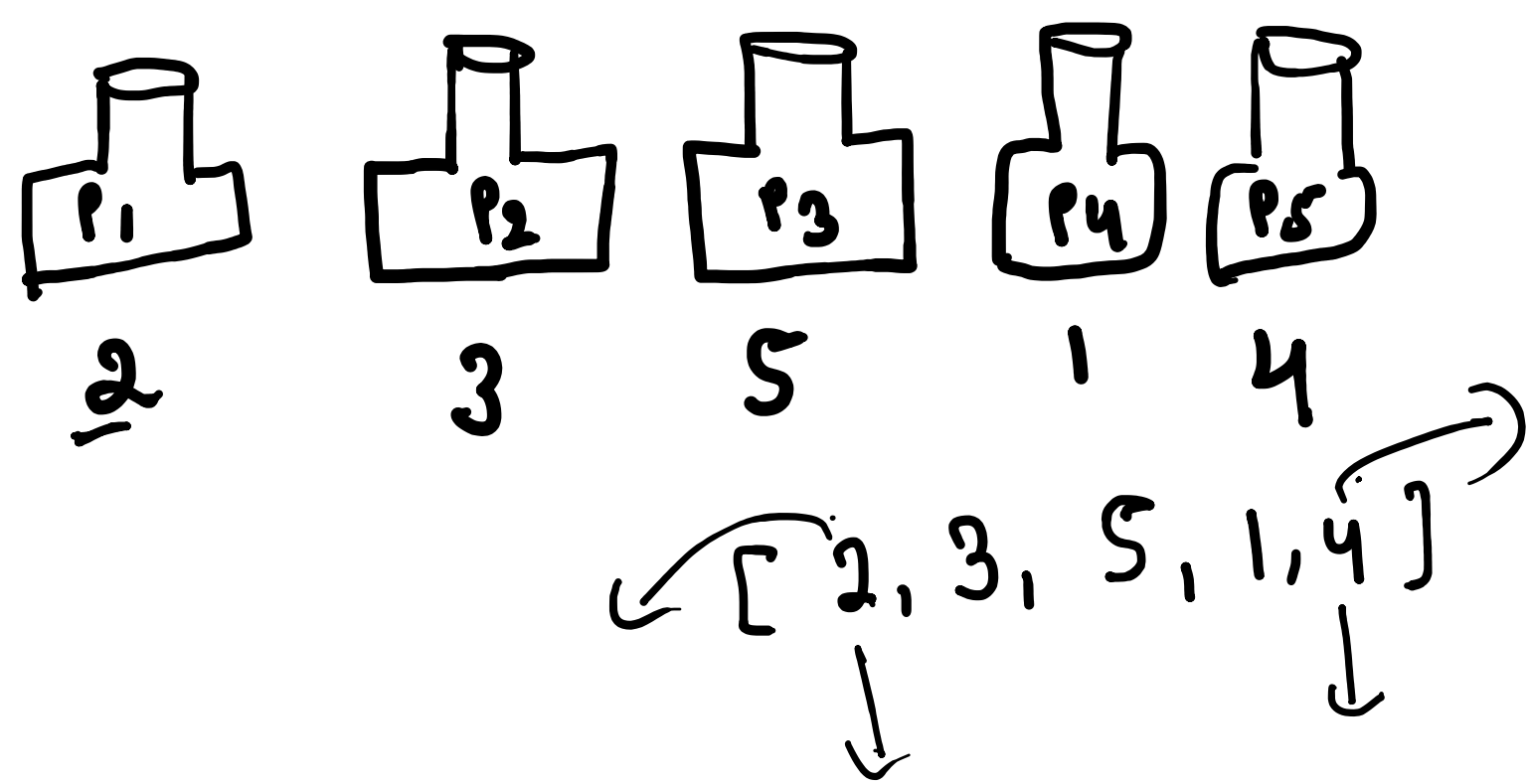
int[] arr = {0, 8, 4, 2, 12, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15};

Handwritten notes show recursive calls for the array [0, 8, 4, 2, 12, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15]. The array is indexed 0 to 15. The recursive calls are shown as follows:

- 0: [0] [8]
- 8: [0] [8]
- 4: [0] [8]
- 2: [0] [8]
- 12: [0] [8]
- 10: [0] [8]
- 6: [0] [8]
- 14: [0] [8]
- 1: [0] [8]
- 9: [0] [8]
- 5: [0] [8]
- 13: [0] [8]
- 3: [0] [8]
- 11: [0] [8]
- 7: [0] [8]
- 15: [0] [8]

The recursive calls are shown as a sequence of calls, with the array being updated at each step. The final result is 15.

Given n wines in a row, with integers denoting the cost of each wine respectively. Each year you can sell the first or the last wine in the row. Let the initial profits from the wines be $P_1, P_2, P_3 \dots P_n$. In the Y^{th} year, the profit from the i^{th} wine will be $Y * P[i]$. The goal is to calculate the maximum profit that can be earned by selling all the wines. Suppose, wine array denotes the initial cost of each wine in the first year. wine[] = [2, 3, 5, 1, 4]



Handwritten calculations for the wine problem:

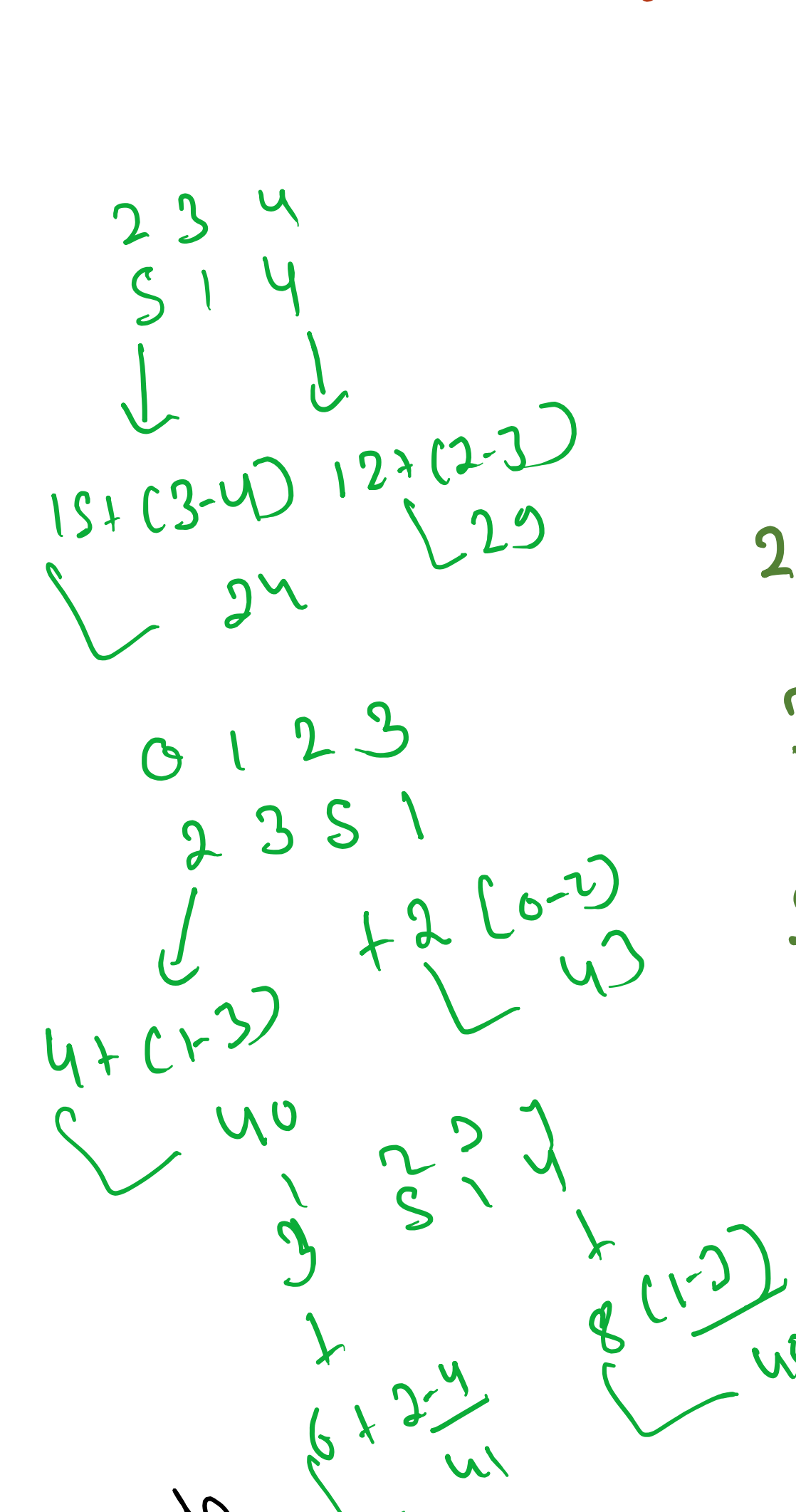
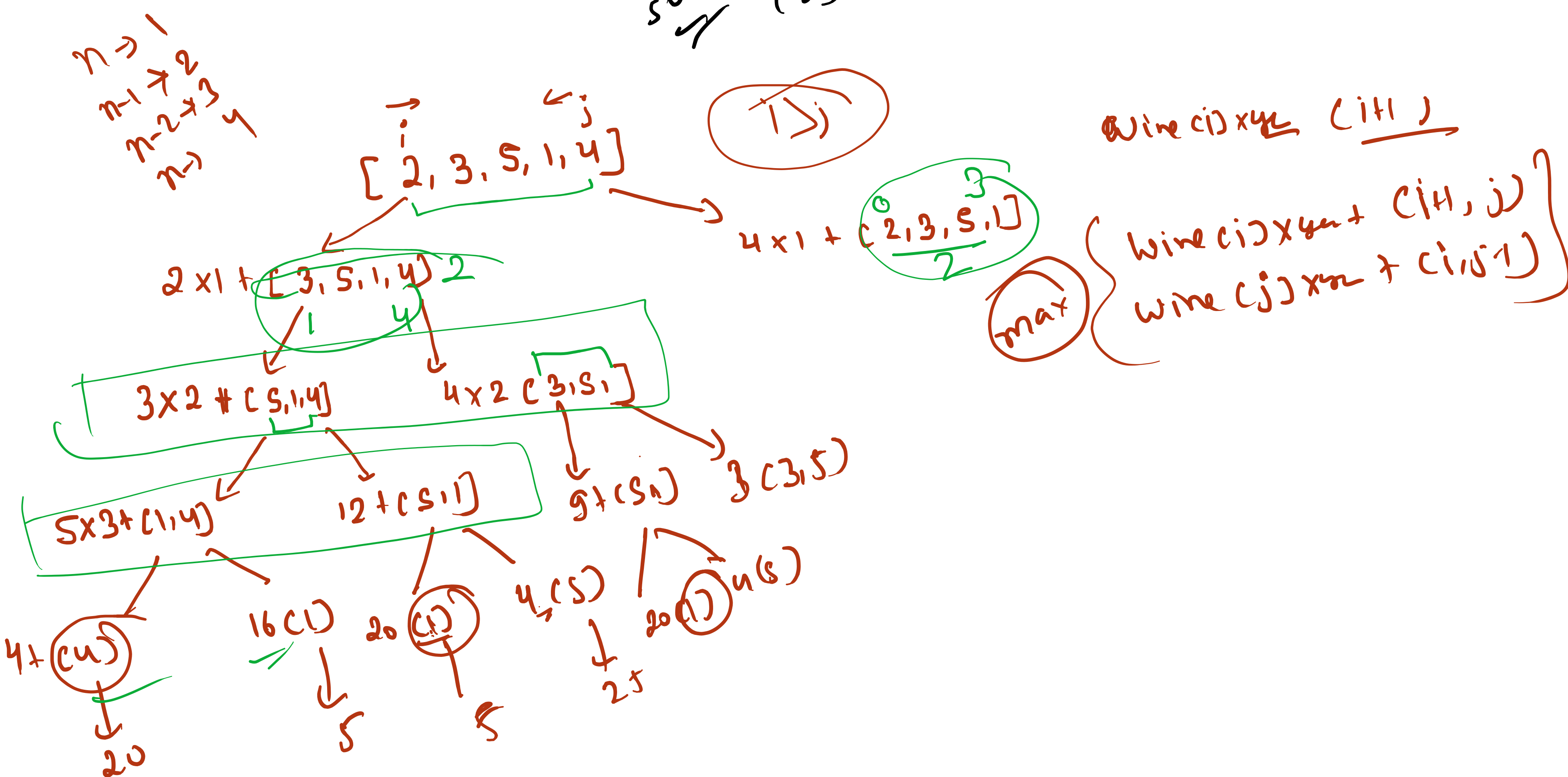
$$2 \times 1 + (3, 5, 1, 4)$$
$$4 \times 1 + (2, 3, 5, 1)$$

Handwritten calculations for the wine problem:

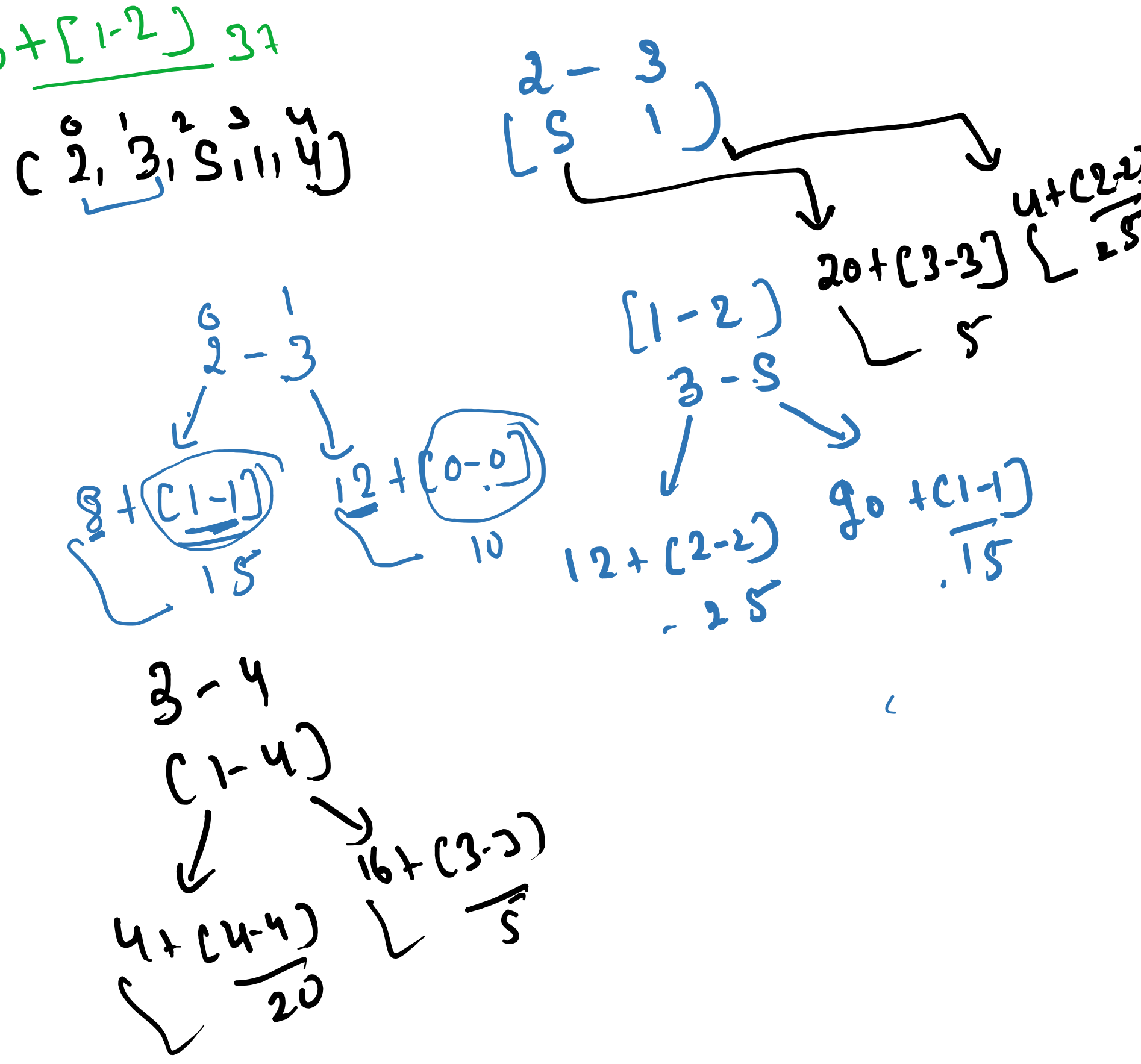
$$2 \times 1 = 2$$
$$3 \times 2 = 6$$
$$4 \times 3 = 12$$
$$1 \times 4 = 4$$
$$5 \times 5 = 25$$
$$\text{Total} = 49$$

Handwritten calculations for the wine problem:

$$2 \times 1 = 2$$
$$4 \times 2 = 8$$
$$3 \times 3 = 9$$
$$5 \times 4 = 20$$
$$\text{Total} = 39$$



	0	1	2	3	4
0	0	23	43	45	50
1		15	37	40	48
2			25	29	41
3				5	24
4					20



	0	1	2	3	4
0	0	1	2	3	4
1		1	2	3	4
2			2	3	4
3				3	4
4					4

Recursion for the wine problem:

Base case: $get = 1$

Recursive case: $get = 1$

Profit: $P[i] = Y * P[i]$

Goal: Calculate the maximum profit that can be earned by selling all the wines.

