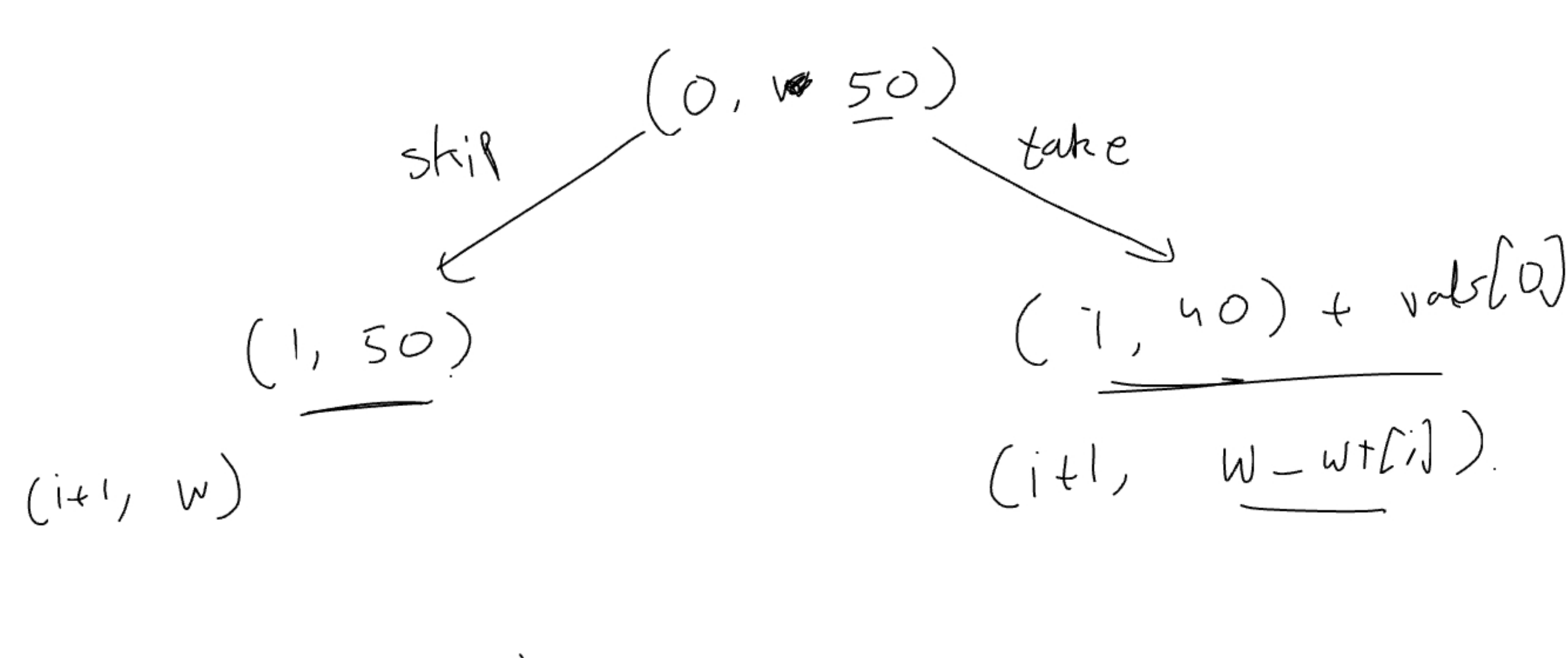
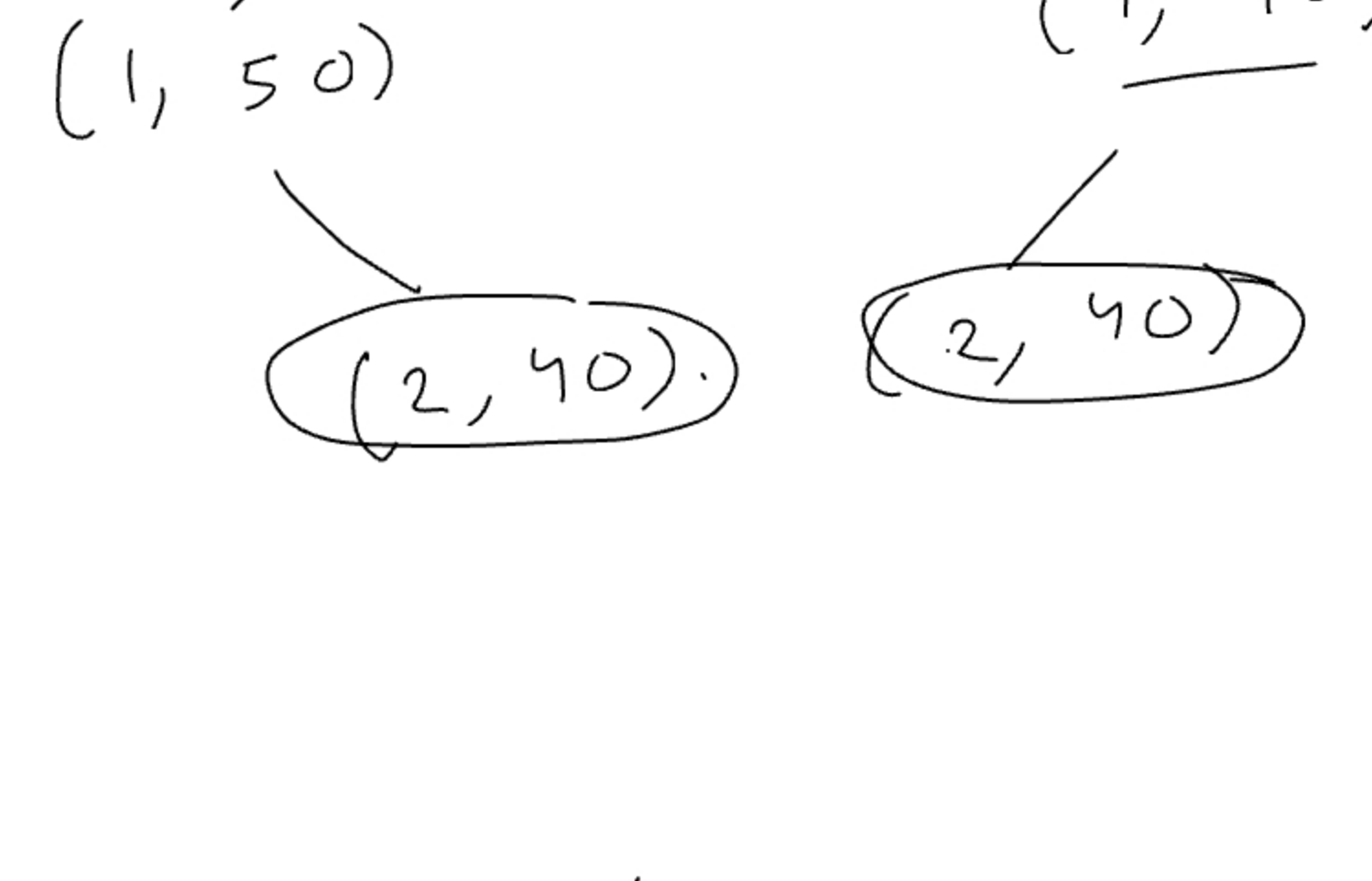


vals: [60, 100, 120]
 wts: [10, 20, 30] $W = 50$



	0	1	2	3	4	5	
vals:	100	120	200	500	70	700	$W = 50$
wts:	10	10	30	40	20	50	



10 20

15 15

```
int getMaxValueUsingBottomUp(vector<int> vals, vector<int> wts, int W) {
    int n = vals.size();
    vector<vector<int>> dp(n + 1, vector<int> (W + 1));

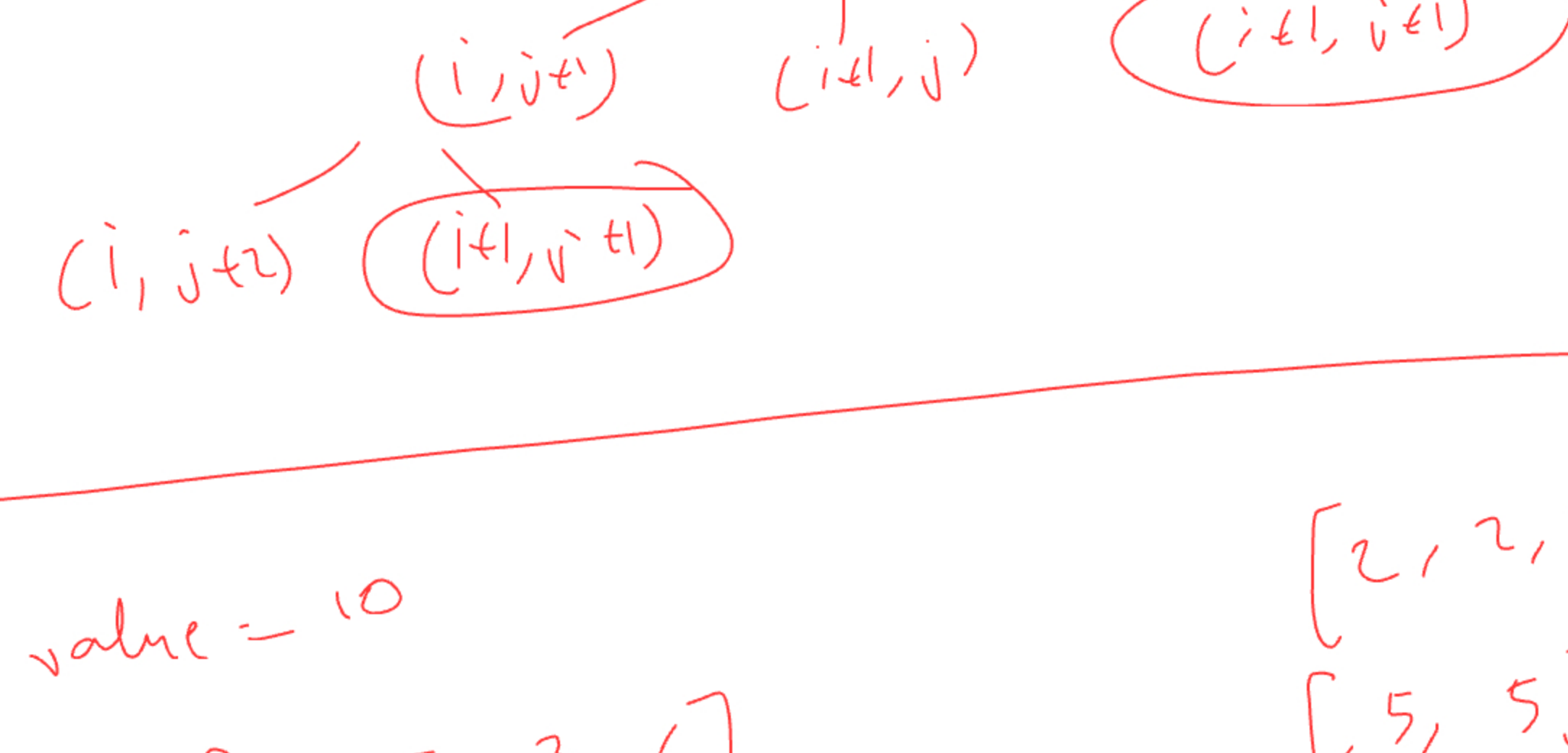
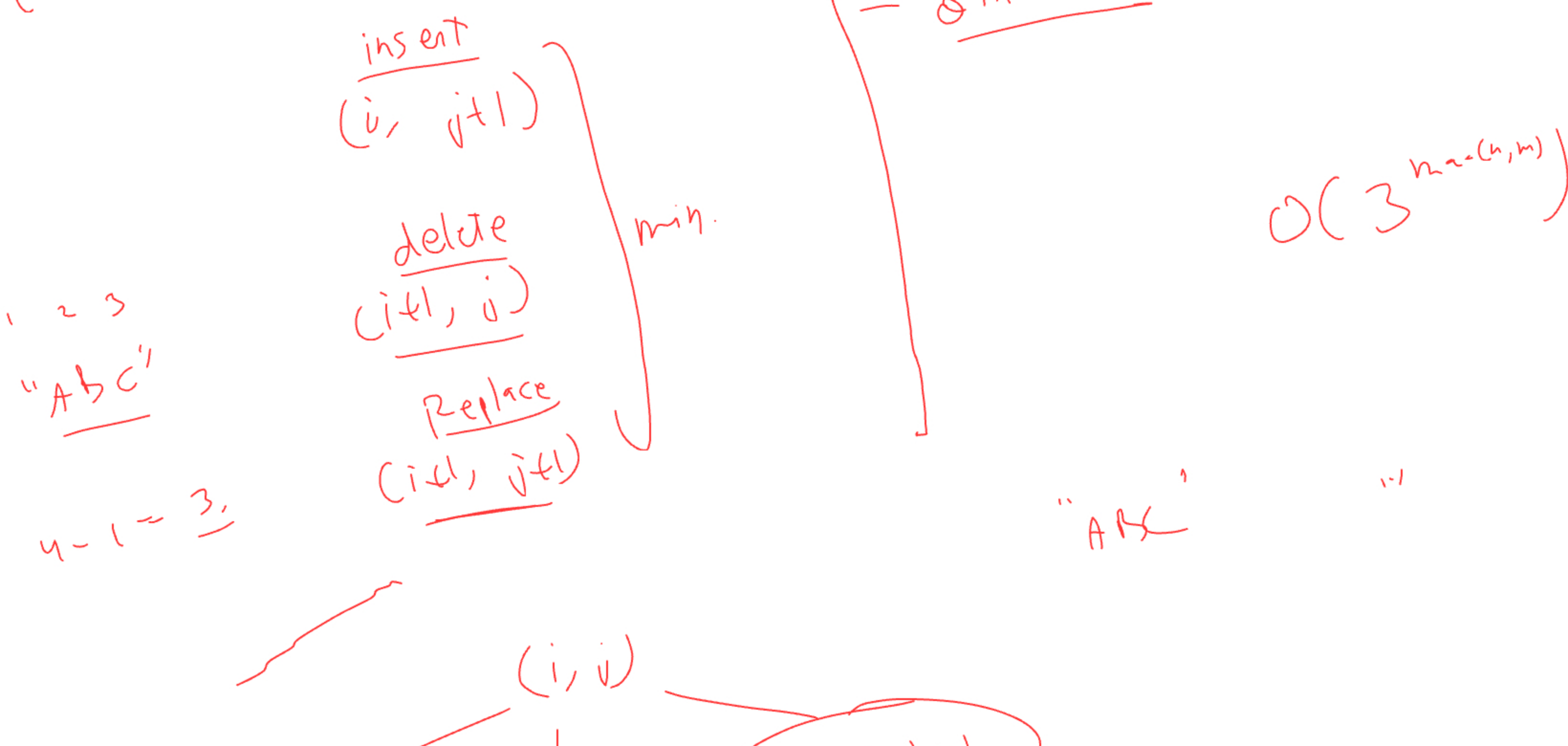
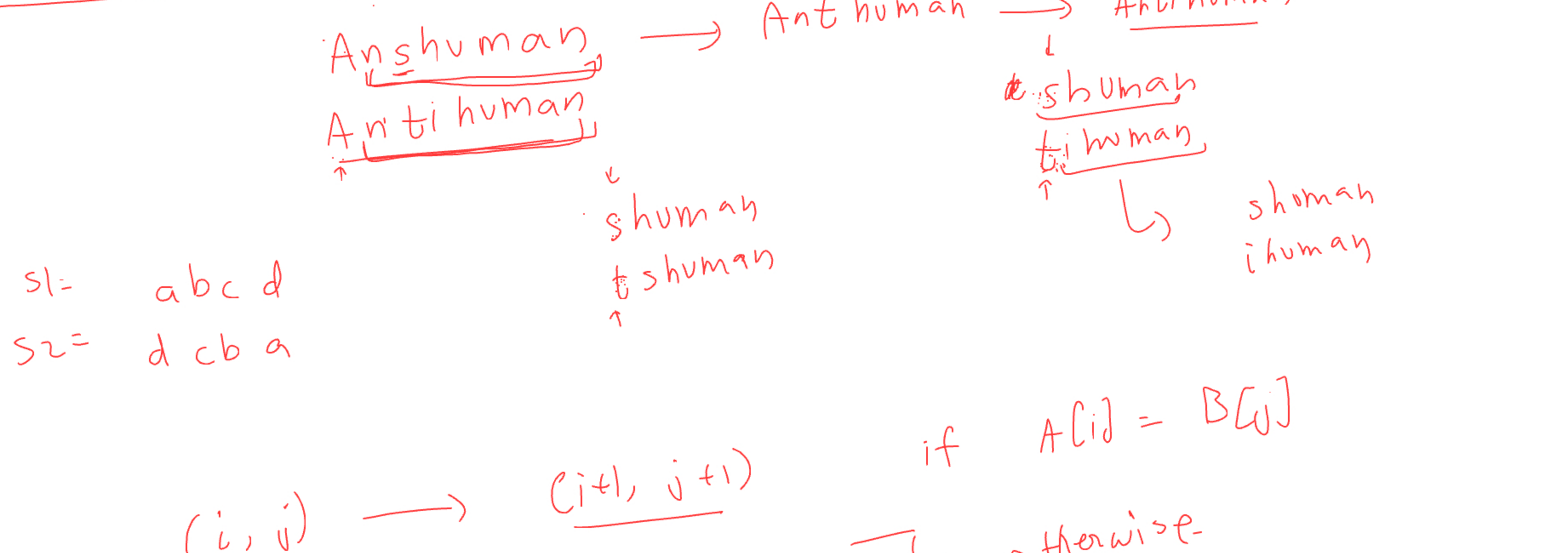
    for (int i = 0; i <= n; i++) {
        dp[i][0] = 0;
    }
    for (int j = 0; j <= W; j++) {
        dp[0][j] = 0;
    }

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= W; j++) {
            dp[i][j] = dp[i - 1][j];
            if (wts[i - 1] <= j) {
                dp[i][j] = max(dp[i][j], vals[i - 1] + dp[i - 1][j - wts[i - 1]]);
            }
        }
    }

    return dp[n][W];
}
```

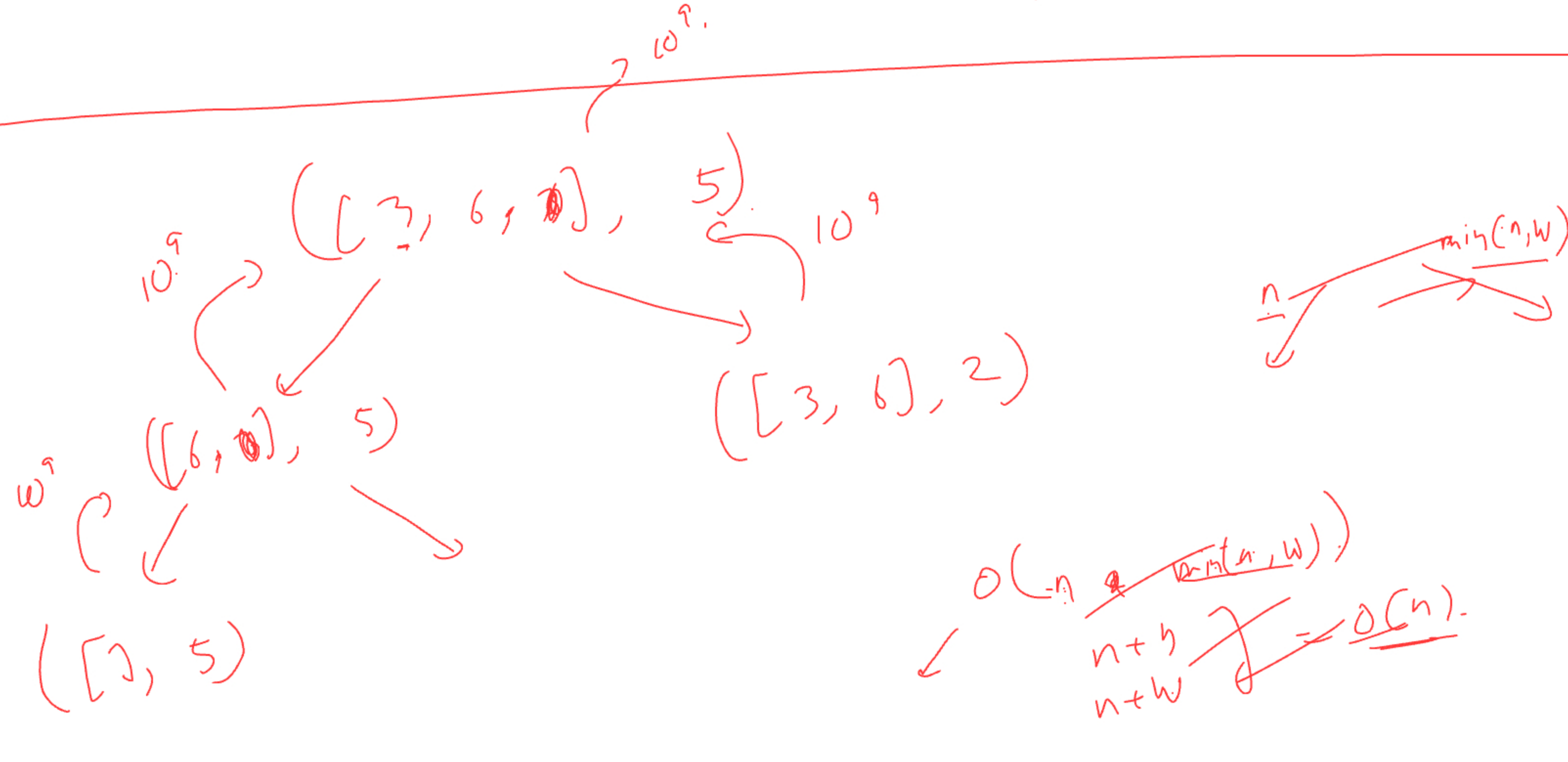
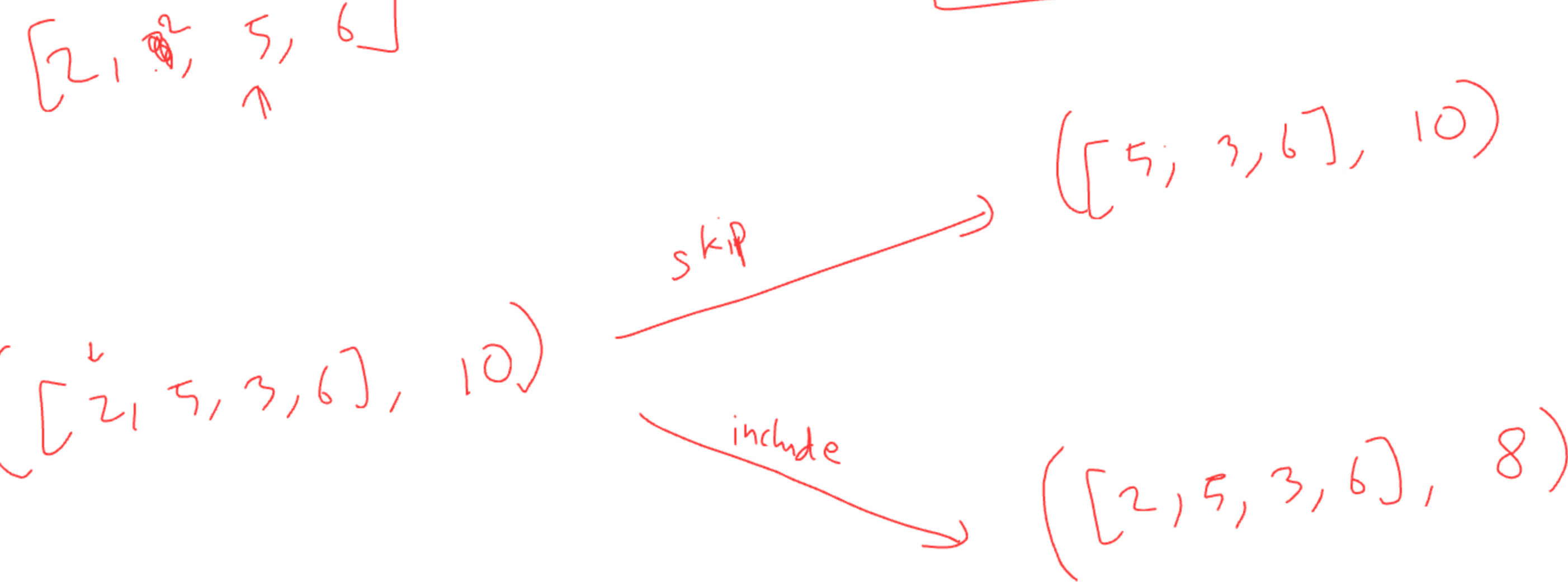
vals: [60, 100, 120]
 wts: [10, 20, 50]
 $W = 50$

	0	1	2	3	...	10	...	20	...	30	...	50
60, 10	0	0	0	0		60		60		120		120
100, 20	0	0	0	0		60	100	100		160		160
120, 30	0	0	0	0		60	100	100		160		160



value = 10
 coins [2, 5, 3, 6]

[2, 2, 2, 2, 2] → 5
 [5, 5] → 2
 [2, 5, 3] → 3



[1, 1, 10, 20] 100

[1, 2, 3, ...]

[5, 3, 7, 10]

104 10 5 15

Optment 7 3 10

[5, 3, 70, 10]

104 10 5 15

Optment 70 3 10

[5, 3, 70, 10]

104 10 5 15

Optment 10 3 10

[5, 3, 5, 70, 10]

104 5 10 5 70

Optment 3 70 10

[5, 3, 5, 70, 10]

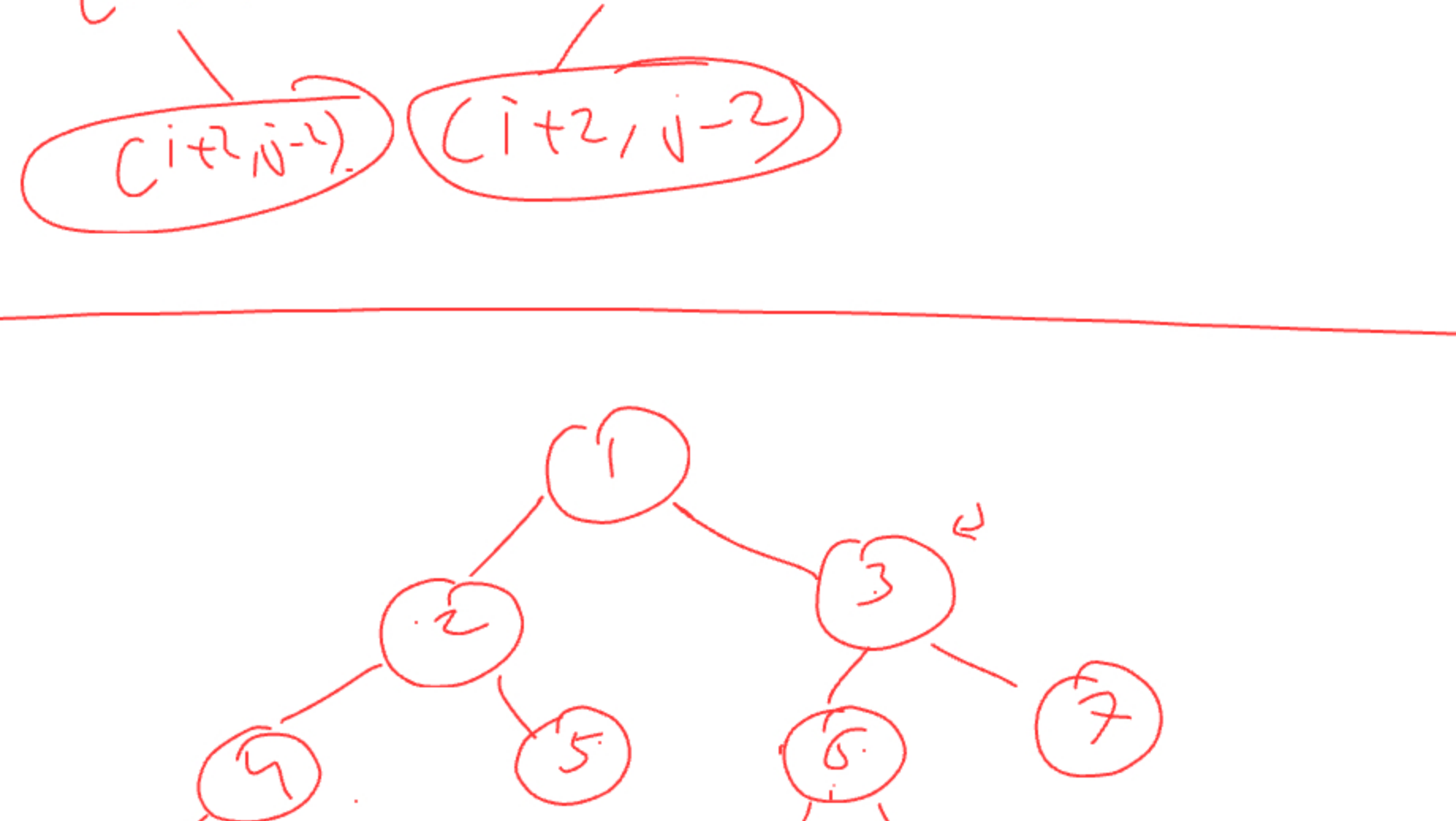
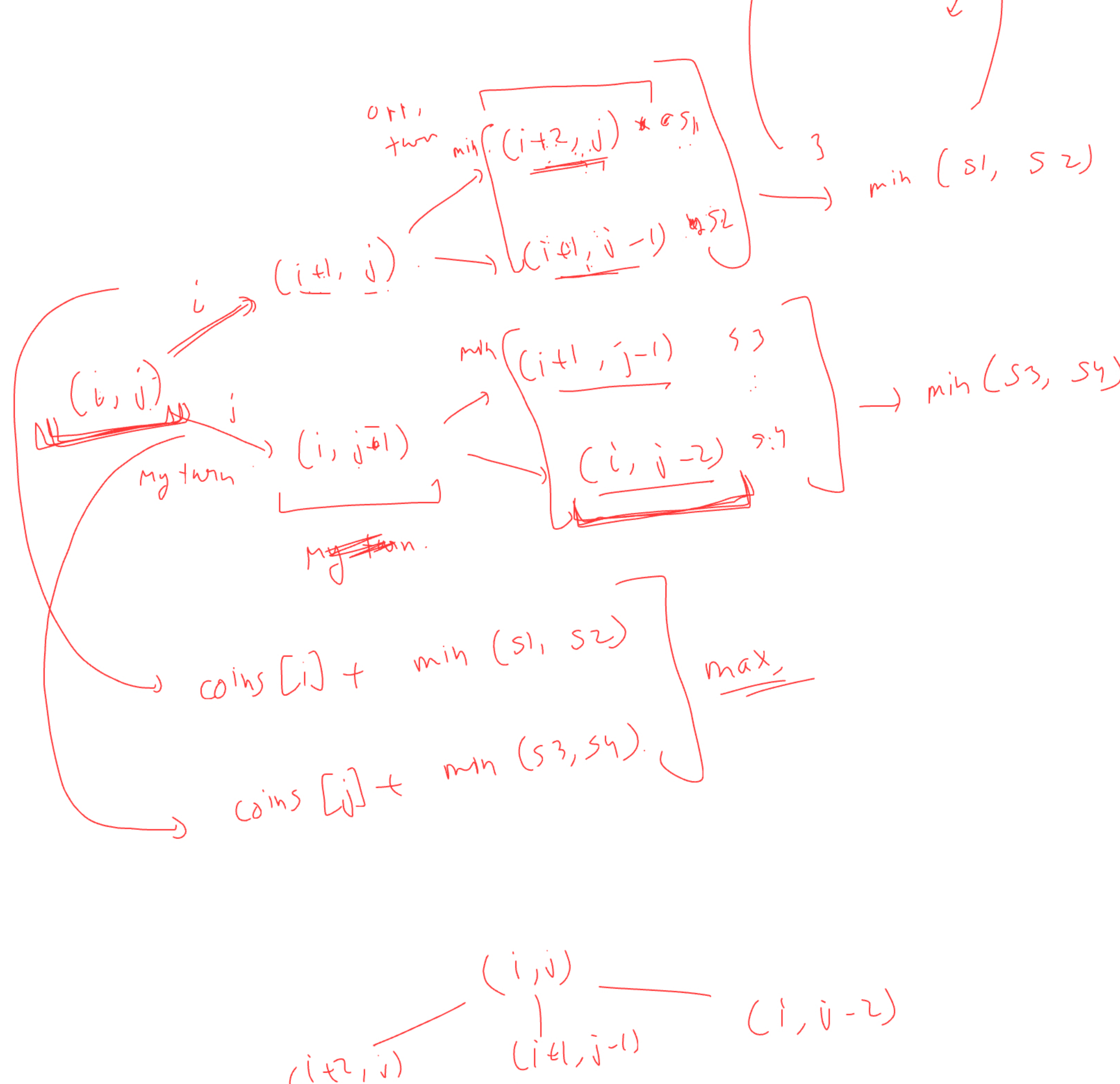
104 5 10 5 70

Optment 3 70 10

[5, 3, 5, 70, 10]

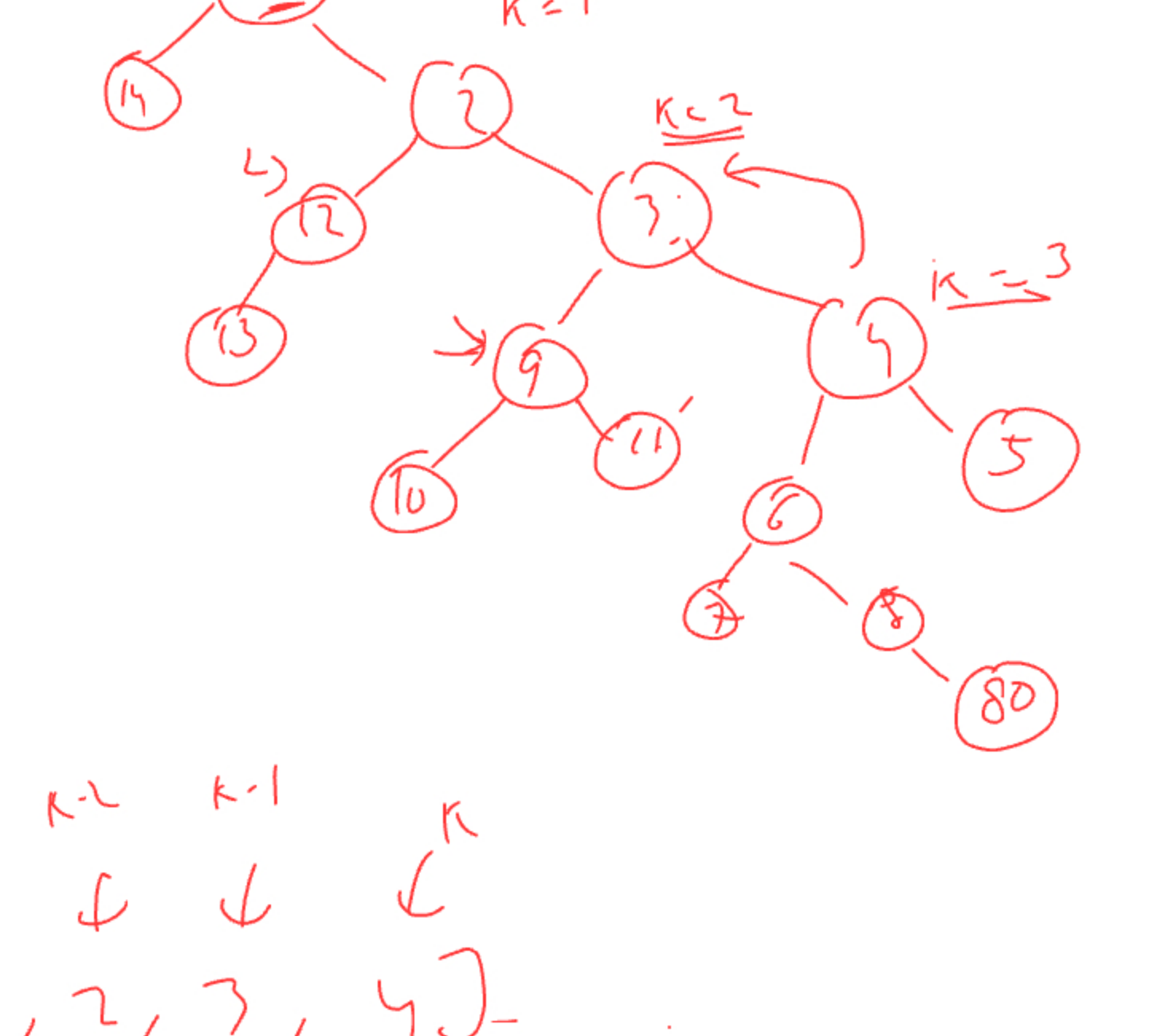
104 5 10 5 70

Optment 3 70 10



for = 3 k = 3

[12, 12, 4, 5]



for = 4 k = 3

[80, 10, 11, 12]

path: [1, 2, 3, 4]

$n - k - 1 \geq 0$