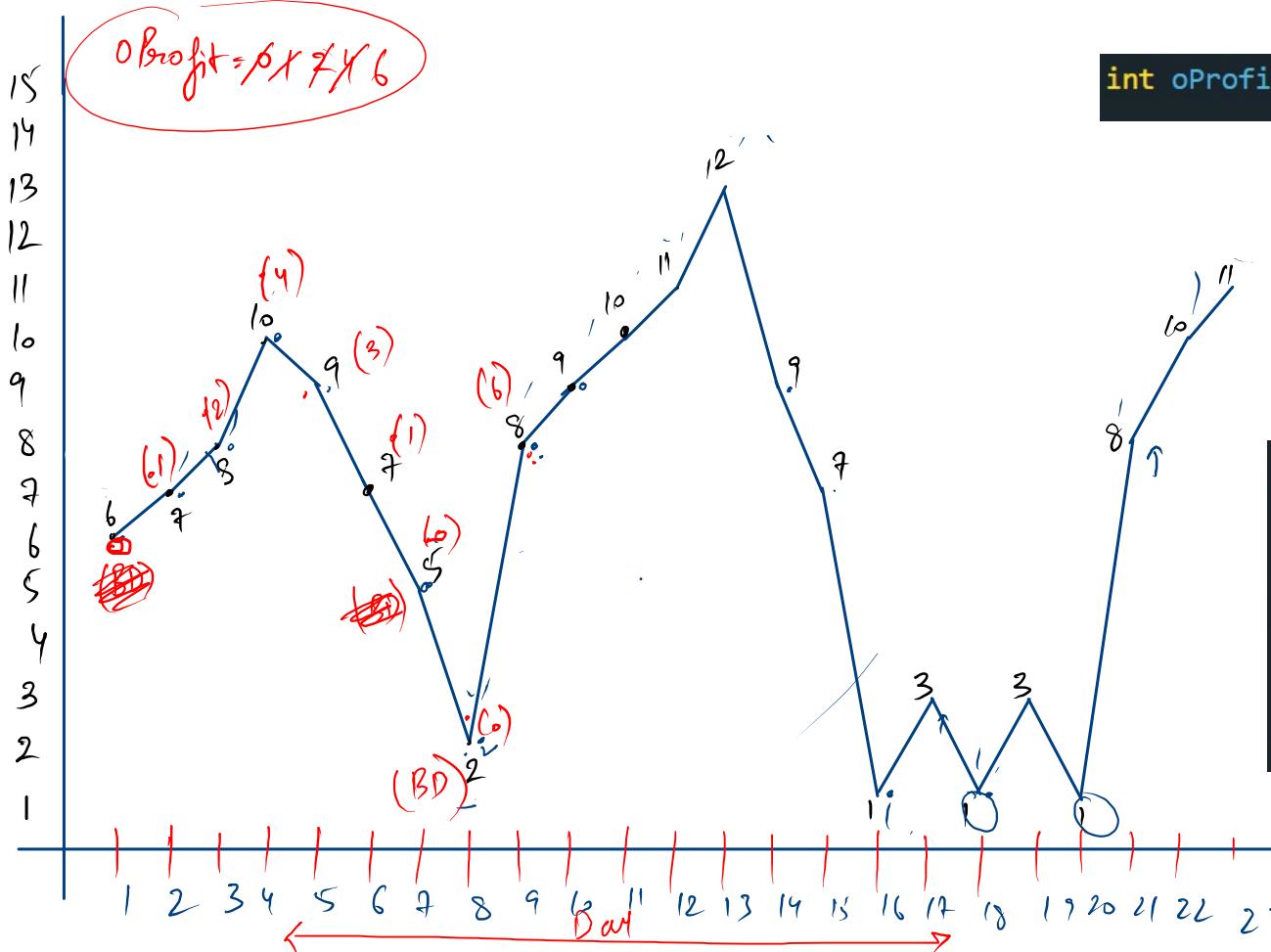


only

1 transaction  
allowed.

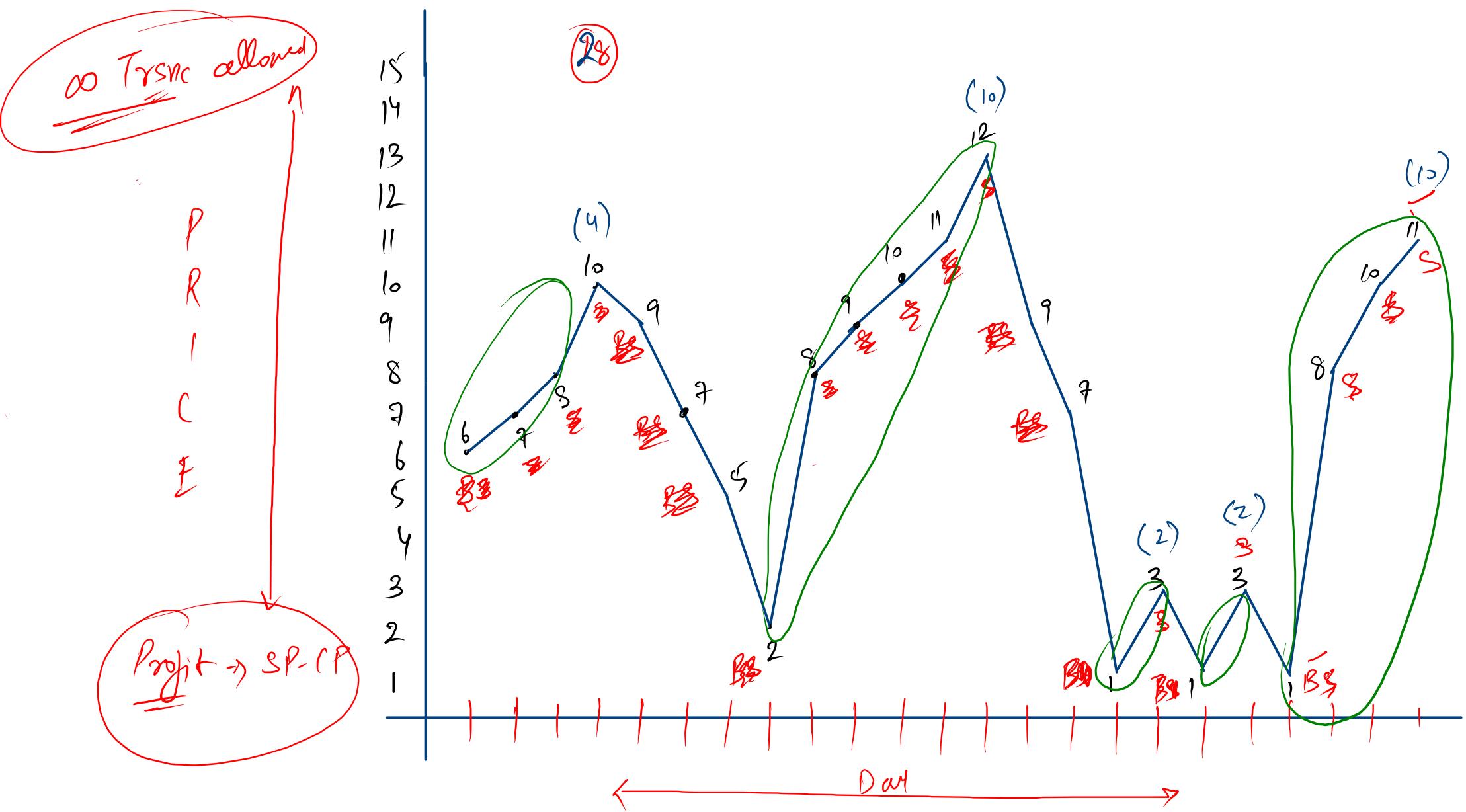
P  
R  
I  
C  
E

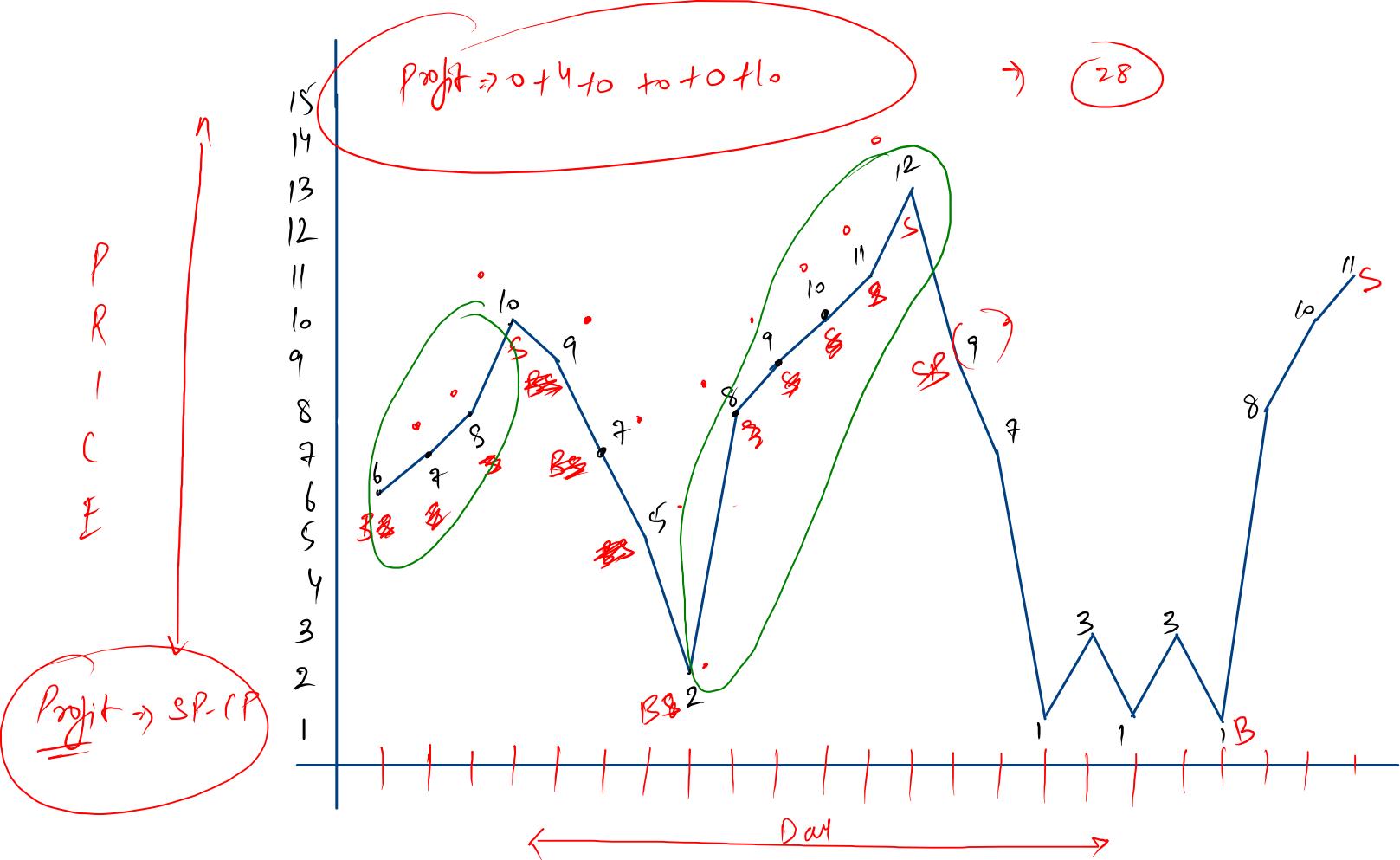
Profit  $\rightarrow SP - CP$   
 $SP \rightarrow \text{Max}$   
 $CP \rightarrow \text{Min}$



```
int oProfit = 0, cProfit = 0, buyDay = 0;
```

```
for(int day = 1 ; day < prices.length ; day++){
    if(prices[day] < prices[buyDay]){
        buyDay = day;
    }
    cProfit = prices[day] - prices[buyDay];
    if(cProfit > oProfit){
        oProfit = cProfit;
    }
}
```





```

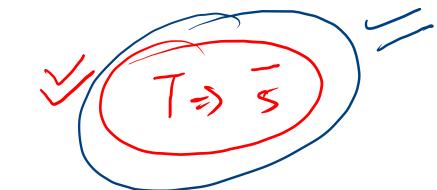
int sp = 0, bp = 0, profit = 0;

for(int day = 1 ; day < prices.length ; day++){
    if(prices[day] >= prices[day-1]){
        sp = day;
    }else{
        profit += prices[sp] - prices[bp];
        sp = day;
        bp = day;
    }
    profit += prices[sp] - prices[bp];
}

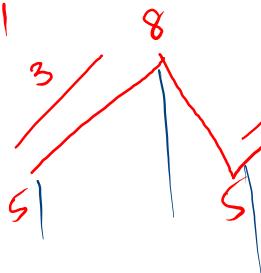
return profit;

```

12  
10 15 17 20 16 18 22 20 22 20 23 25 3



⑨ -10 → -1



6 ↗ ↘ 11

∞ + Trsc-fee

BS

↙ What ↘  
↙ Why ↗ ↘

↙ ↘ 1 Transaction allowed ↗ ↘ 1

↙ ↘ 2 Transaction allowed ↗ ↘ -1

12

10 15 17 20 16 18 22 20 22 20 23 25

(3)

Type 23

Days	Price
0	10
1	15
2	17
→ 3	20
→ 4	16
→ 5	(18)
→ 6	22
→ 7	20
→ 8	22
→ 9	20
→ 10	23
→ 11	25

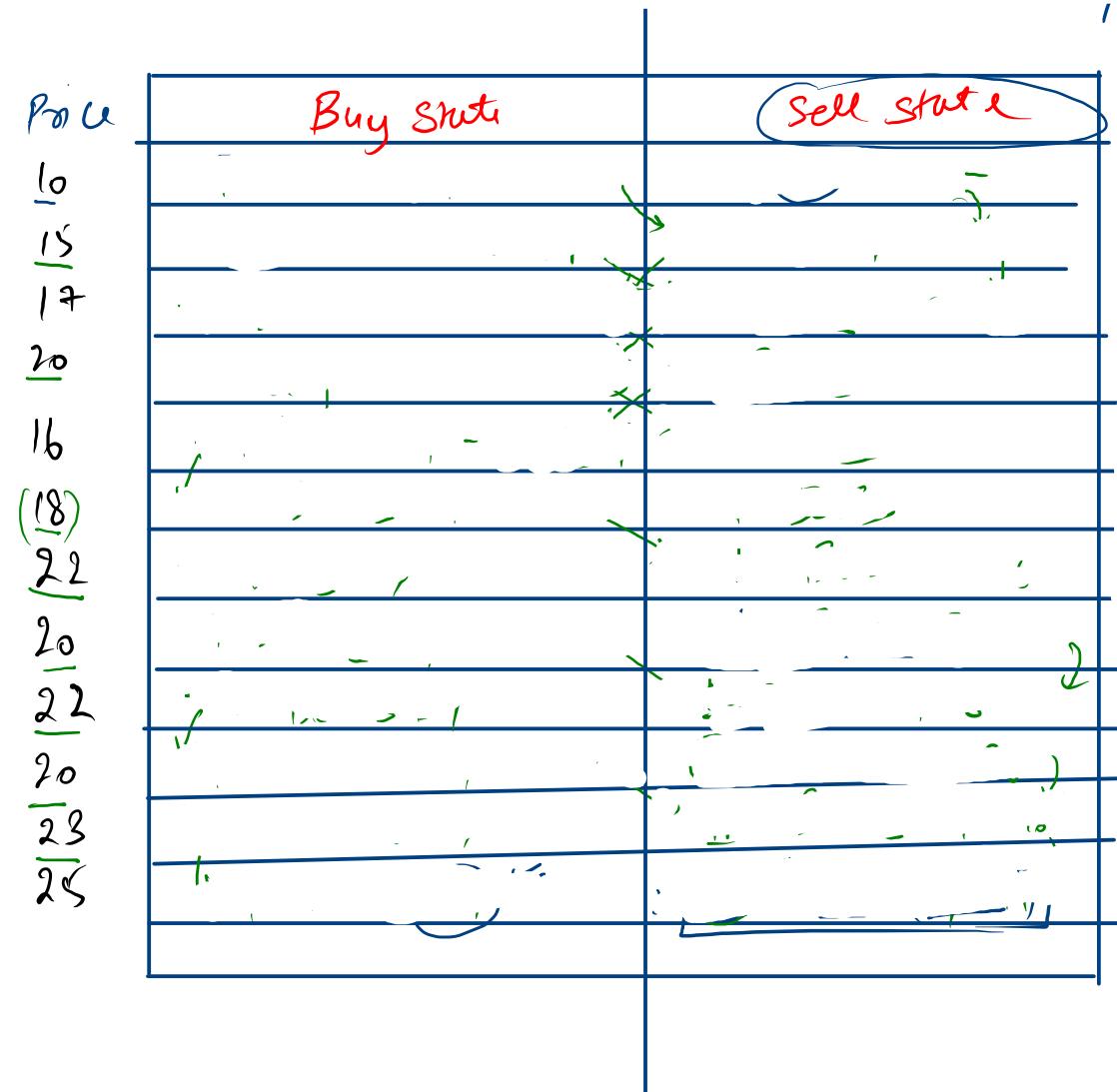
[        ]

 $B_S \Rightarrow -3$ 

↑

Buy state	Sell state
-10 $B_0$	0
-10 $B_0$	2 $B_0 S_1$
-10 $B_0$	4 $B_0 S_2$
-10 $B_0$	7 $B_0 S_3$
-9 $B_0 S_3 B_1$	7 $B_0 S_3$
-9 $B_0 S_3 B_1$	2 $B_0 S_3$
-9 $B_0 S_3 B_1$	10 $B_0 S_3 B_1 S_6$
-9 $B_0 S_3 B_1$	10 $B_0 S_3 B_1 S_6$
-9 $B_0 S_3 B_1$	10 $B_0 S_3 B_1 S_8$
-9 $B_0 S_3 B_1$	(10) $B_0 S_3 B_1 S_8$
-9 $B_0 S_3 B_1$	11 $B_0 S_3 B_1 S_{10}$
-9 $B_0 S_3 B_1$	(12) $B_0 S_3 (B_1 S_{11})$

(13) ≈



$T \rightarrow S$

Price

10

15

17

20

16

(18)

22

20

22

20

23

25

Buy State

-10

$$0 - 15 = -15, (-10)$$

$$(-10), -15$$

$$-16, (-10)$$

$$(-9), -10$$

$$-10, (-10)$$

$$(B)$$

$$-10, (-10)$$

$$-10, (-10)$$

$$-10, (-10)$$

$$-10, (-10)$$

$$-10, (-10)$$

$$(B)$$

Sell State

0

2

4

7

7

3

(S)

2

0

2

0

2

0

2

(S)

```
int oBSP = 0, oSSP = 0;
```

```
for(int day = 0; day < prices.length; day++) {
```

```
    if(day == 0) {
```

```
        oBSP = -prices[0];
```

```
    } else {
```

```
        int nBSP = Math.max(oSSP - prices[day], oBSP);
```

```
        int nSSP = Math.max(oSSP, prices[day] - tFee + oBSP);
```

```
        oBSP = nBSP;
```

```
        oSSP = nSSP;
```

```
}
```

```
return oSSP;
```

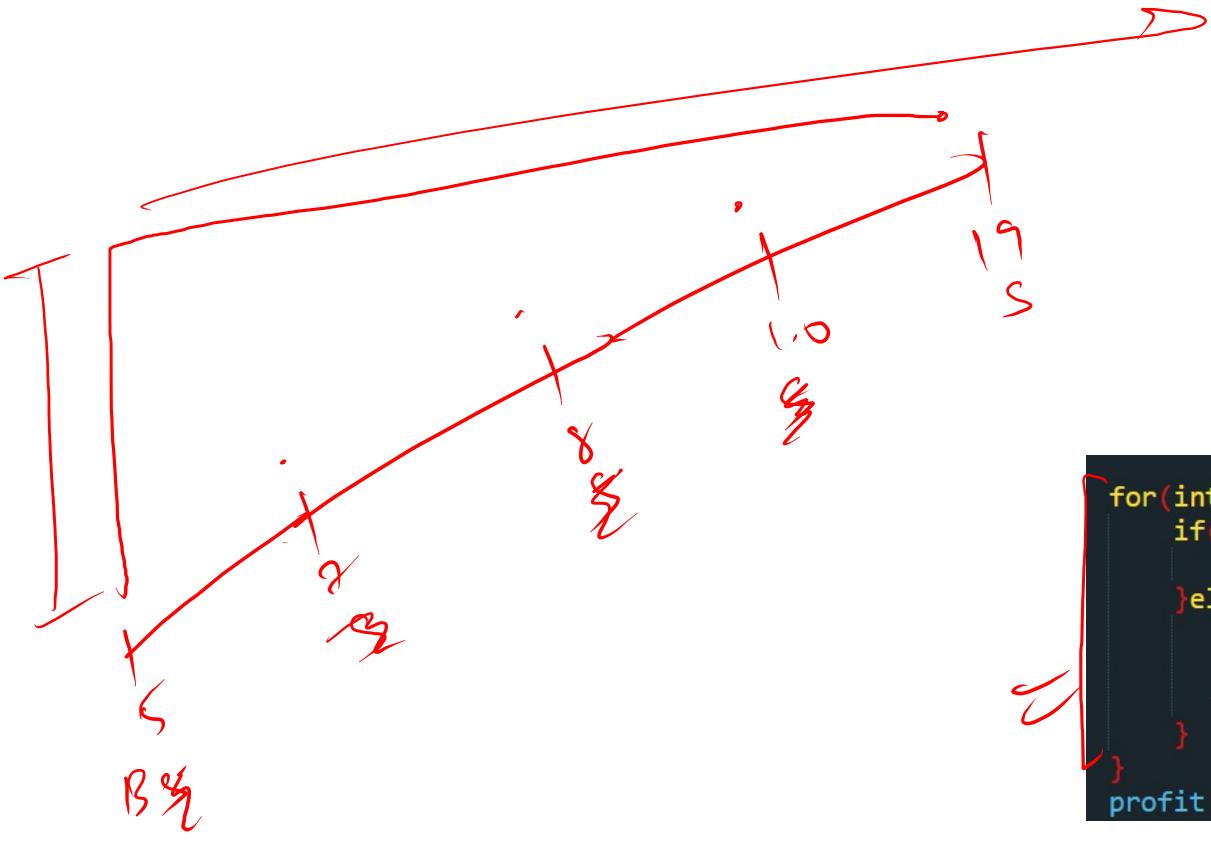
$$\begin{array}{r} 16 \\ -3 \\ \hline 13 \end{array}$$

$$OBSP = 0 - 10$$

$$nBSP = -9$$

$$oSSP = 0 / 4 / 2$$

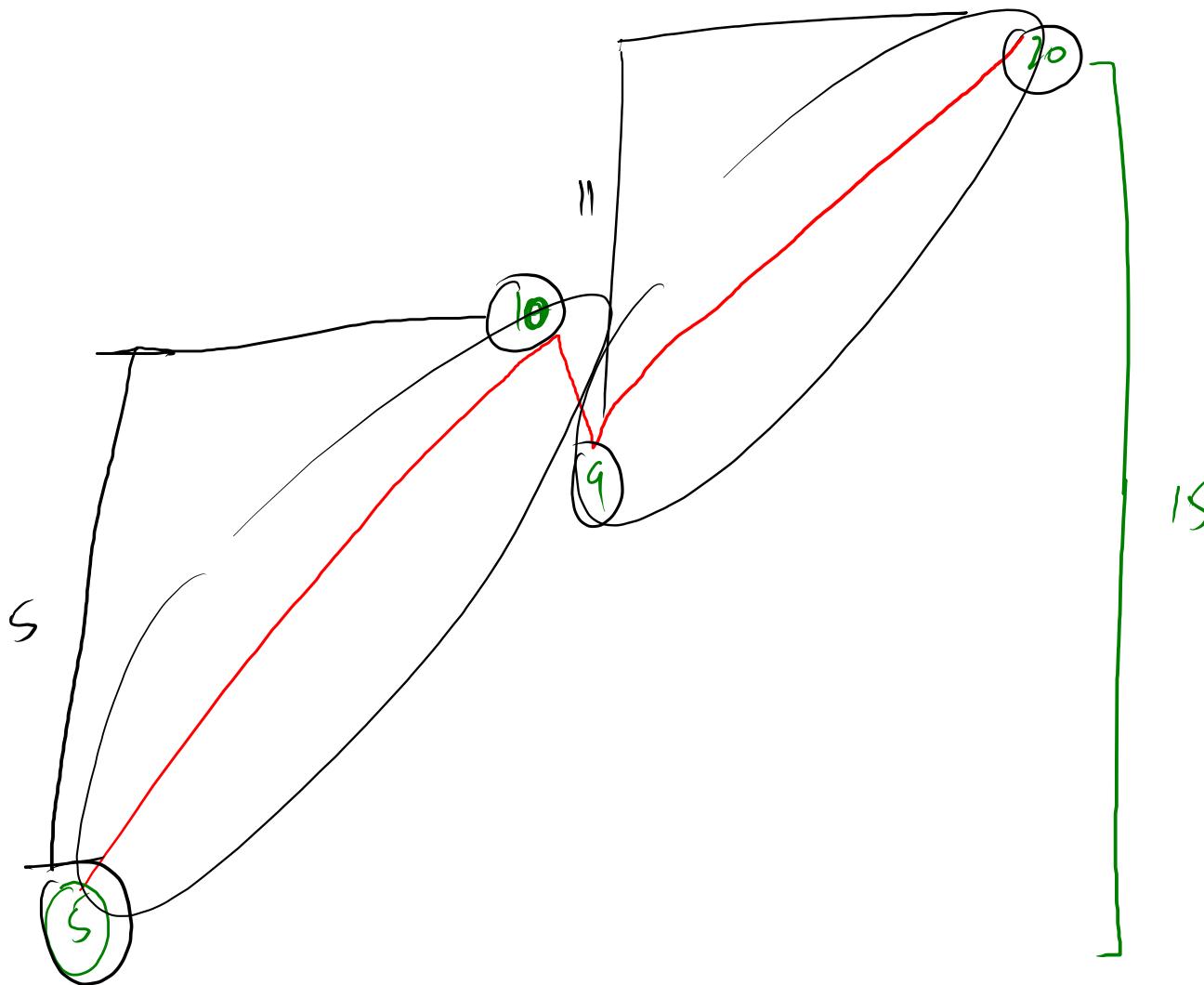
$$nSSP =$$



```
for(int day = 1 ; day < prices.length ; day++){
    if(prices[day] >= prices[day-1]){
        sp = day;
    }else{
        profit += prices[sp] - prices[bp];
        sp = day;
        bp = day;
    }
}
profit += prices[sp] - prices[bp];
```

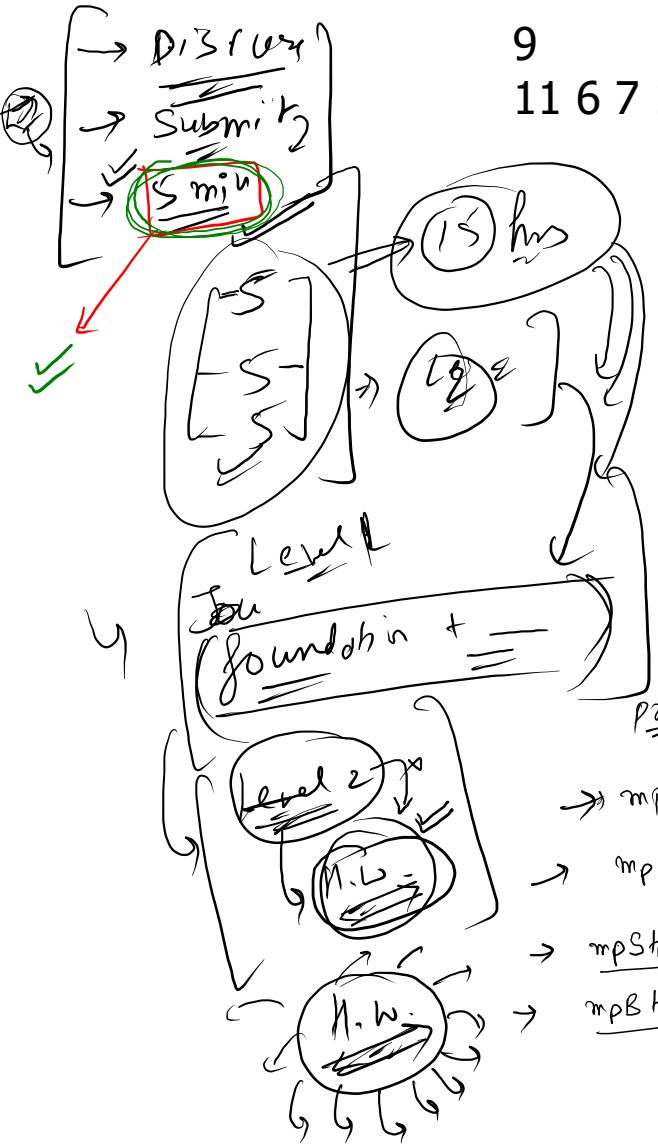
①  $\Rightarrow 20 - 5 = 15$

②  $\Rightarrow$



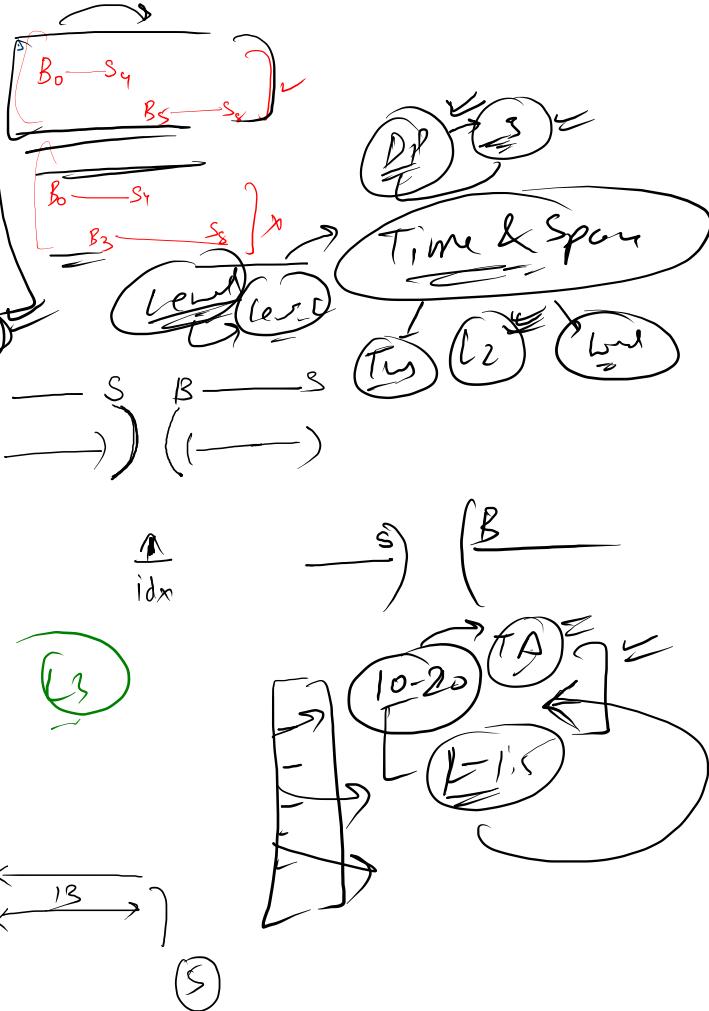
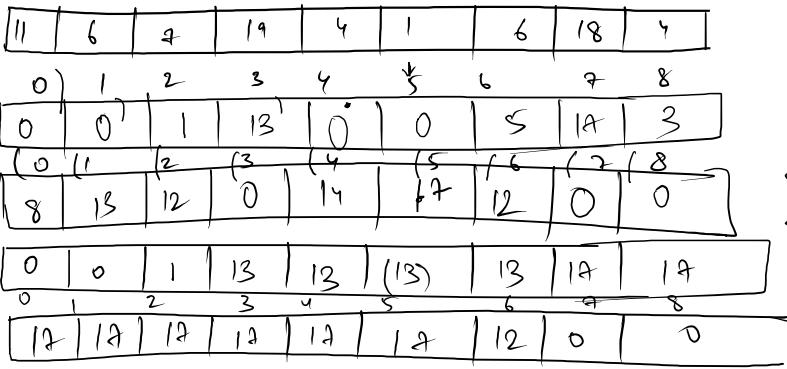
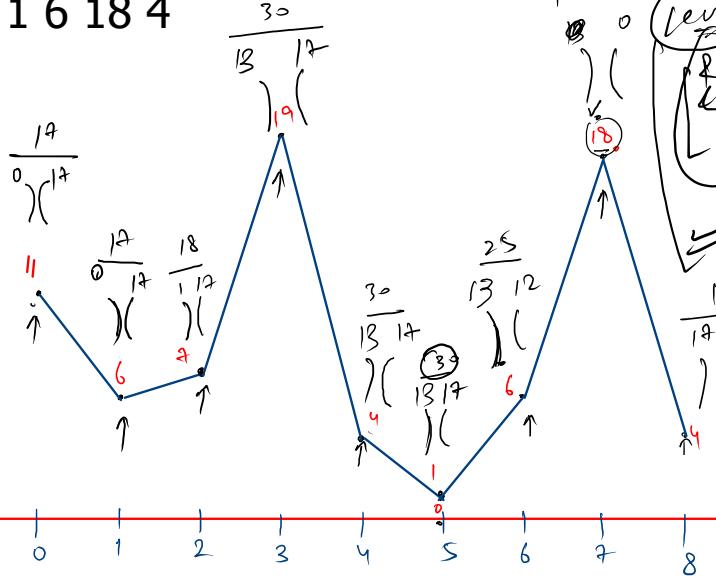
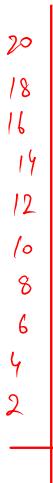
	BSP	SSP	<u>CSP</u>
0 →	10	-10 $B_0$	0 -      0 -
1 →	15	-10 $B_0$	5 $B_0 S_1$ 0 -
2 →	17	-10 $B_0$	? $B_0 S_2$ $S B_0 S_1 l_2$
3 →	90	-10      . <del><math>B_0</math></del>	
4	96		
5	18		
6	22		
7	20		
8	22		
9	23		
10	25		

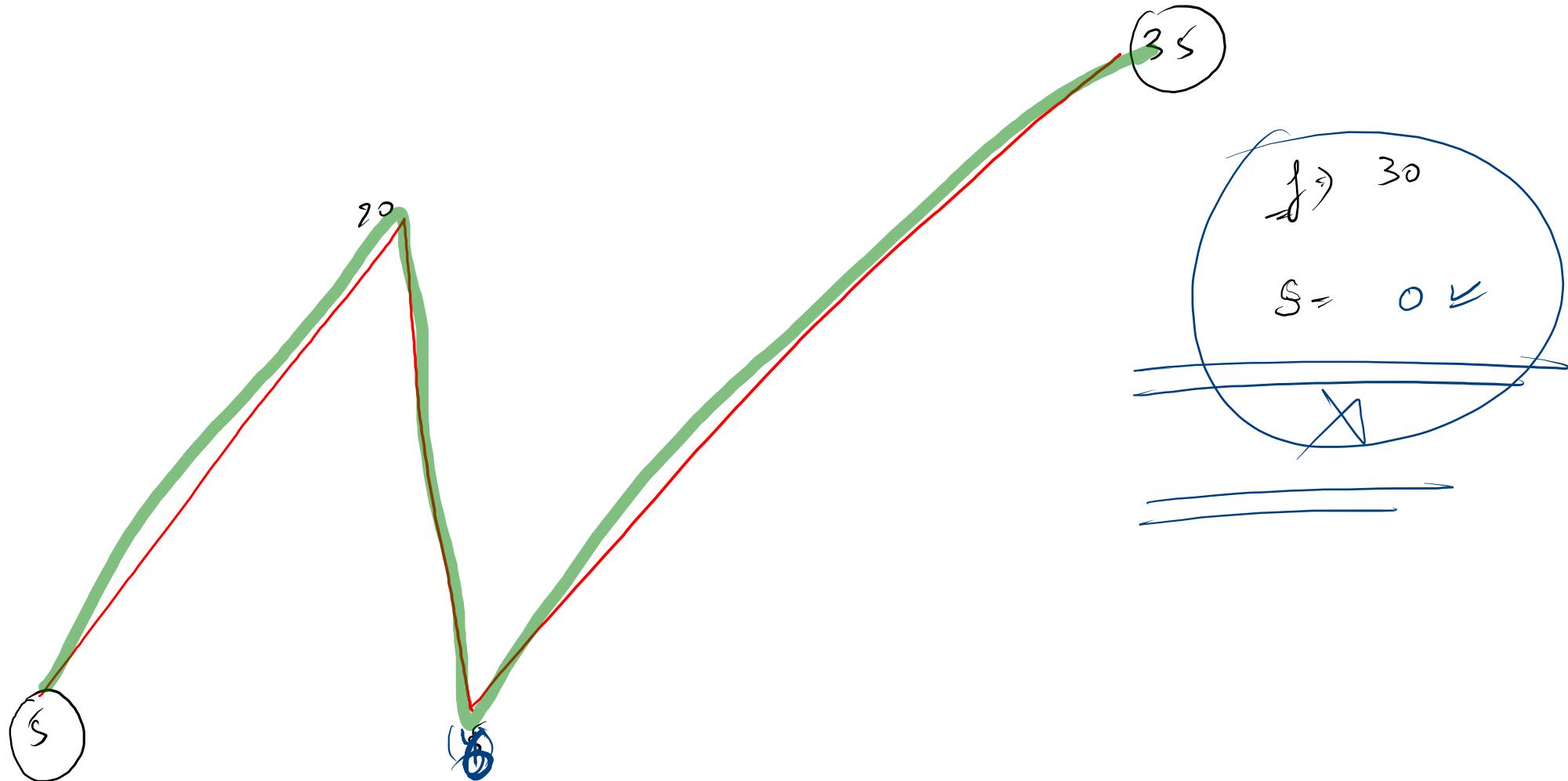




9

11 6 7 19 4 1 6 18 4





16

19

