121. Best Time to Buy and Sell Stock

link

普通的图表法

这个是找到当前节点之前最小的那个值(当前最大盈利), 跟res对比

```
public int maxProfit(int[] prices) {
    if(prices.length < 1) return 0;
    int res = 0;
    int low = prices[0];
    for(int price : prices){
        if(price < low)
            low = price;
        else
            res = Math.max(res,(price - low));
    }
    return res;
}
```

122. Best Time to Buy and Sell Stock II

link

一模一样, 只是盈利可以累加

```
public int maxProfit(int[] prices) {
    if(prices.length == 0) return 0;
    int max = 0;
    int low = prices[0];
    for(int i = 1; i < prices.length; ++i){
        if(prices[i] < low){
            low = prices[i];
        }else{
            max += prices[i] - low;
```

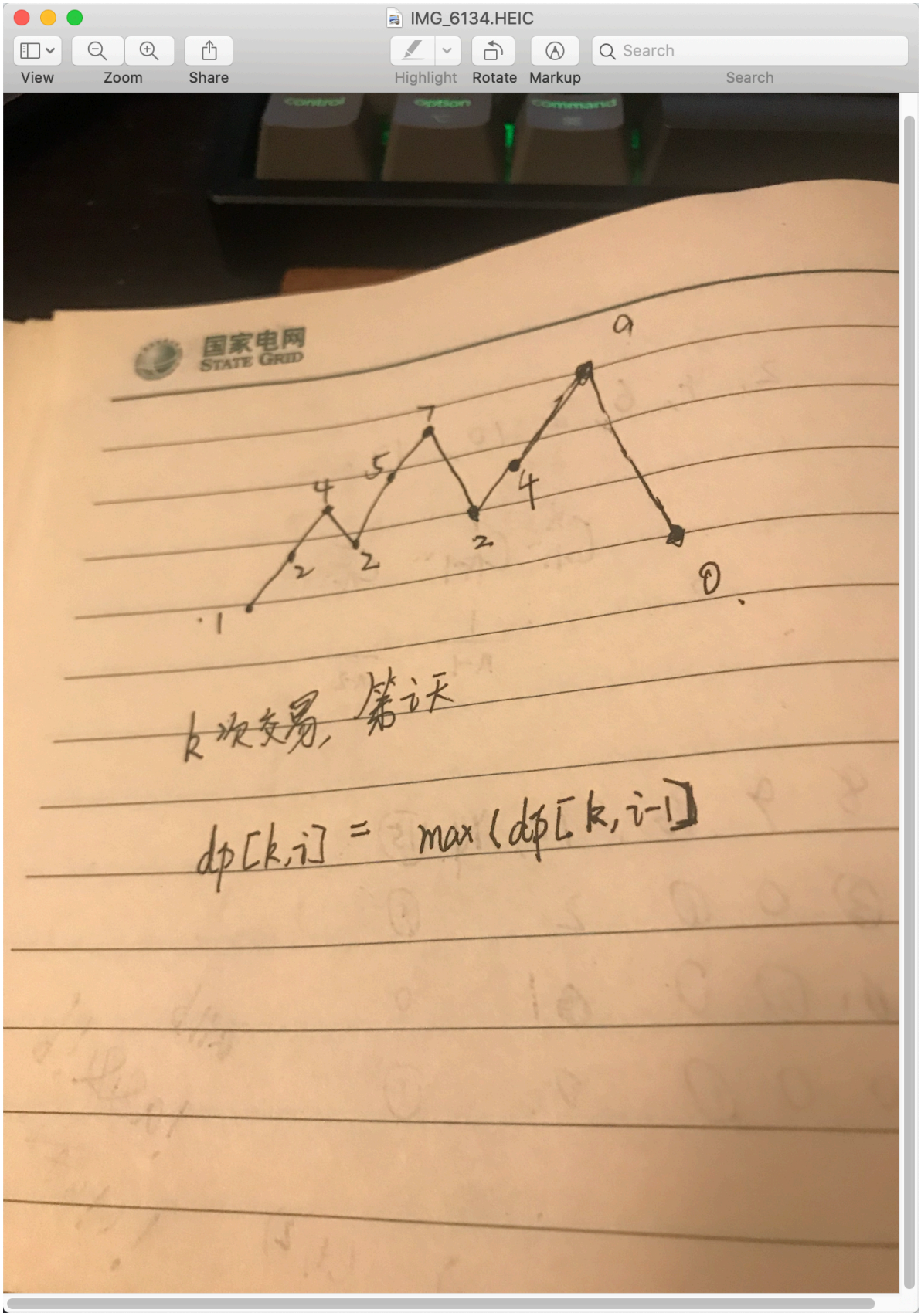
```
                low = prices[i];
            }
        }
        return max;
    }
```

123. Best Time to Buy and Sell Stock III

link

explain link

第i 天k次购买的最大值, 是等于第i-1天k次购买的最大值或者第j天购买了第k-1次, 然后加上(i-j)天的盈利和第j-1天k-1的值


IMG_6134.HEIC
View
Zoom
Share
Highlight
Rotate
Markup
Search
Search
国家电网
STATE GRID
k次交易，第i天
dp[k,i] = max(dp[k,i-1]

例如我开始写的错误的只考虑了局部

1,2,4,2,5,7,2,4,9,0如图，是在7的时候卖第一次，9的时候卖第二次.

9的时候的最大值= Math.max(第八天进行第二次交易，第j天进行第第一次交易 max(j) + price[i] - price[j])

```
//开始错误的写法
    public int maxProfit(int[] prices) {
        if(prices.length == 0) return 0;
        int[] profit = new int[2];
        int low = prices[0];
        int max = 0;
        for(int i = 1; i < prices.length; ++i){
            if(prices[i] >= low){
                max += prices[i] - low;
                if(i + 1 == prices.length || prices[i + 1] < prices[i]){
                    updateProfit(profit, max);
                    max = 0;
                }
            }
            low = prices[i];
        }
        return profit[0] + profit[1];

    }
    public void updateProfit(int[] profit, int value){
        if(value > profit[0]){
            if(value > profit[1]){
                profit[0] = profit[1];
                profit[1] = value;
            }else{
                profit[0] = value;
            }
        }
    }
```