

Elaine_DWL

博客园 首页 新随笔 联系 订阅 管理

【leetcode】股票买卖系列总结

股票买卖系列总结

股票买卖系列的题目在面试中还是比较经典的，这里对这一系列做一些简单的总结。

1. 只允许买卖一次

假设股票价格序列为 (3, 5, 7, 3, 8, 1)

动态规划。整个过程中的行为选择有3种，买/卖/无操作。

用 $dp[i]$ 表示第 i 天的行为是“卖出”时，能得到的最大收益。显然，我们固定了卖出的时间，只要在这个时间点之前的时间中选择股票价格比最小的时候买入，就可以确定 $dp[i]$ 的值。

所以我们可以遍历价格序列，cur_min不断更新以记录当前时间点之前的股票最低价格，所以 $dp[i] = \max(0, prices[i] - cur_min)$ ，最后所有 $dp[i]$ 中的最大值就是只进行一次买卖能得到的最大利润。可以在求 $dp[i]$ 的同时用一个变量来记录其最大值。

因为 $dp[i]$ 只用到一次，所以没必要开一个数组专门来存储。

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        // 只允许最多一次交易 求最大收益
        // 记录当前最小 cur_min dp[i]表示在第i天卖出的最大收益
        if(prices.size()<=1) return 0;
        int cur_min = prices[0], res = 0;
        for(int i=1; i<prices.size(); i++){
            res = max(prices[i]-cur_min, res); // dp[i] = prices[i] - cur_min;
            cur_min = min(cur_min, prices[i]);
        }
    }
};
```

公告

昵称： Elaine_DWL
园龄： 5年3个月
粉丝： 0
关注： 12
[+加关注](#)

< 2020年9月 >						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

搜索

找找看

谷歌搜索

我的标签

```
        return res;
    }
};
```

2. 不限制买卖次数

假设股票价格序列为 `(1, 2, 3, 4, 5)` 。其实这是一个特殊情况，我们可以在第0天买入，然后在最后一天卖出。此时最大收益就是4。

我们可以在第0天买入，第一天卖出。第一天买入，第二天卖出。。。这样的结果和上面是一样的，最大收益都是4。

所以，对于比较常规的情况，比如说 `(7, 6, 3, 4, 5)` ，在第一天和第二天是不能买入的，因为，之后找不到比当前价格更大的数了。另外的话，可以在3的时候买入，4的时候卖出，然后4又买入，5卖出。最后最大收益是2。

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        // 不限制股票的买卖次数
        // 只要是遇到比当前价格更高的就卖掉
        if(prices.size() <=1) return 0;
        int buy = INT_MAX, profit = 0;
        for(int i=0; i<prices.size(); i++){
            if(prices[i]<=buy) buy = prices[i];
            else{
                profit += prices[i]-buy;
                buy = prices[i];
            }
        }
        return profit;
    }
};
```

可以将上面代码进行简化：

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        if(prices.size() <=1) return 0;
        int buy = INT_MAX, profit = 0;
        for(int i=0; i<prices.size(); i++){
            if(prices[i] > buy) profit += prices[i] - buy;// 只有当当前价格大于假设买入的价格时，才进行卖出
            buy = prices[i];// 每次都在当前进行“假设”买入
        }
        return profit;
    }
};
```

3. 最多只能买卖2次

*buy1*表示在第*i*天进行第一次买入时可以获得的最大收益

*buy2*表示在第*i*天进行第二次买入时可以获得的最大收益

*sell1*表示在第*i*天进行第一次卖出时可以获得的最大收益

*sell2*表示在第*i*天进行第二次卖出时可以获得的最大收益

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        // 最多只能买卖2次 求最大收益
        // buy1 sell1 buy2 sell2 分别表示当前天如果是第一次买/卖、第二次买/卖时的最大收益
        int buy1 = INT_MIN, buy2 = INT_MIN, sell1 = 0, sell2 = 0;
```

刷题(24)
算法模板(13)
C++(7)
机器学习(5)
leetcode(4)
原理推导(4)
深度学习(4)
目标检测(2)
python(2)
环境配置(2)
更多

随笔档案
2019年7月(8)
2018年10月(4)
2018年9月(3)
2018年8月(2)
2018年3月(1)
2017年12月(1)
2017年11月(2)

```
        for(int price:prices){
            sell2 = max(sell2, buy2+price);
            buy2 = max(buy2, sell1-price);
            sell1 = max(sell1, buy1+price);
            buy1 = max(buy1, -price);
        }
        return sell2;
    }
};
```

4. 有冷冻期不限制买卖次数

$buy[i]$ 表示第*i*天之前最后一次行为是buy时，最大的收益

$sell[i]$ 表示第*i*天之前最后一次行为是sell时，最大的收益

$rest[i]$ 表示第*i*天之前最后一次行为是冷冻rest时，最大的收益

$buy[i] = \max(buy[i - 1], rest[i - 1] - prices[i])$ ，max(第i天冷冻, 第i天是卖出)

$sell[i] = \max(sell[i - 1], buy[i - 1] + prices[i])$ ，max(第i天冷冻, 第i天是买入)

$rest[i] = \max(rest[i - 1], buy[i - 1], sell[i - 1])$

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        // 有冷冻期 不限制买卖次数 但是卖完股票后有一天的冷冻期才能再接着买
        int n = prices.size();
        if(n<=1) return 0;
        vector<int> buy(n, INT_MIN), sell(n, INT_MIN), rest(n, INT_MIN);
        buy[0] = -prices[0];
        sell[0] = 0;
        rest[0] = 0;
        for(int i=1; i<n; i++){
            buy[i] = max(buy[i-1], rest[i-1]-prices[i]);
            sell[i] = max(sell[i-1], buy[i-1]+prices[i]);
            rest[i] = max(rest[i-1], max(buy[i-1], sell[i-1]));
        }
        return max(buy[n-1], max(sell[n-1], rest[n-1]));
    }
};
```

标签: leetcode, 刷题

好文要顶

关注我

收藏该文

Elaine_DWL
关注 - 12
粉丝 - 0

+加关注

0

0

« 上一篇：【深度学习系列】DNN中梯度消失和梯度爆炸的原因推导
» 下一篇：【刷题】字节跳动2018算法笔试第二题

posted @ 2019-07-17 18:29 Elaine_DWL 阅读(1365) 评论(0) 编辑 收藏

2017年9月(1)
2017年7月(4)
2017年5月(7)
2017年4月(7)
2017年3月(11)
2017年2月(3)
2015年10月(1)

最新评论

1. Re:先序非递归建立二叉树

@ Elaine_DWL主函数里面的T是指针类型，直接传过去感觉应该对，但是运行出来是错的，加上取地址就对了，不知道为什么。 ...

--橙子123

2. Re:先序非递归建立二叉树

@ 橙子123不用啊 你看看BiTree的定义...

--Elaine_DWL

3. Re:先序非递归建立二叉树

函数形参里面的BiTree R要改成 BiTree&R吧？

--橙子123

阅读排行榜

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#) ， [访问](#) [网站首页](#)。

- 【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【推荐】为自己发“声” —— 声网RTC征文大赛在园子里征稿

【推荐】未知数的距离，毫秒间的传递，声网与你实时互动

【推荐】了不起的开发者，挡不住的华为，园子里的品牌专区

【推荐】SSL证书一站式服务，上海CA权威认证

【推荐】阿里技术3年大合辑免费电子书一键下载

1. C++字符串和数字格式转化（使用printf()和scanf()函数）(5819)

2. 传教士与野人过河问题(A*搜索 C++) (3266)

3. 2016蓝桥杯C++A组第六题 寒假作业【暴力搜索】(2284)

4. 先序非递归建立二叉树(1982)

5. 2016蓝桥杯C++A组第七题 剪邮票【DFS】(1714)

评论排行榜

1. 先序非递归建立二叉树(3)

推荐排行榜

1. C++字符串和数字格式转化（使用printf()和scanf()函数）(1)

相关博文：

- [leetcode-2](#)

· [leetcode-10](#)

· [leetcode-11](#)

· [leetcode-14](#)

· [leetcode-3](#)

» [更多推荐...](#)

最新 IT 新闻：

- [927国际聋人日，科技互联网公司怎样让人「听清」](#)

· [威马的夏天](#)

· [小米全自动智能门锁体验：开门全自动，少拧一下很省心](#)

· [2020北京车展硬核满满，看小度车载玩转智能潮流！](#)

· [个人恩怨？微软获 GPT-3 独家授权，马斯克：OpenAI 不 open 了](#)

» [更多新闻...](#)

