

项目报告

2021-2022第2学期

`

**课 程 名： 算法设计与程序设计竞赛**

**题 目： 买股票的最佳时期**

**专业班级： 计算机科学与技术34班**

**任课教师： 曾 智**

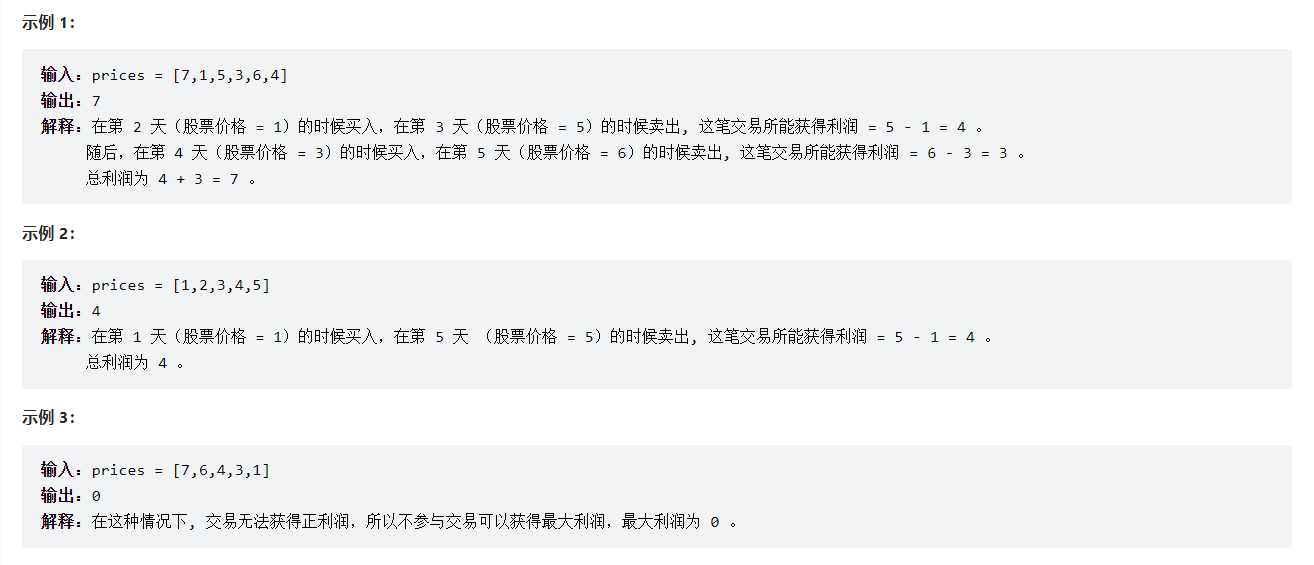
|  |  |  |
| --- | --- | --- |
| 姓名 | 学 号 | 备 注 |
| 杨鸿强 | 2020051615308 |  |
| 杨昌松 | 2020051615325 |  |
| 张诗雨 | 2020051615288 |  |
| 袁雪琴 | 2020051615230 |  |

**2022年6月15日**

1. **问题描述**

给定一个数组，它的第i个元素是一支给定股票第i天的价格。设计一个算法来计算你所能获取的最大利润。你可以尽可能地完成更多的交易（多次买卖）。

**注意：你不能同时参与多笔交易（你必须在再次购买前出售掉之前的股票）。**



1. **dp数组以及下标的含义**

考虑到“不能同时参与多笔交易”，因此每天交易结束后只可能只有这一支股票或者没有股票。

·dp[i][0] 表示第i天交易完后手里没有股票的最大利润。

·dp[i][1] 表示第i天交易完后手里持有一支股票的最大利润。

**注意：这里说的是“持有”，“持有”不代表就是当天“买入”！也有可能是昨天就买入了，今天保持持有的状态。**

1. **确定递推公式和子结构**

子结构:

dp[i][j]：i表示第i天交易完后的最大收益，j可取值为0和1，分别表示当天不持有股票和当天持有股票。

考虑dp[i][0]的转移方程，如果这一天交易完后手里没有股票，那么可能的转移状态为前一天已经没有股票，即dp[i−1][0]，或者前一天结束的时候手里持有一支股票，即dp[i−1][1]，这时候我们要将其卖出，并获得prices[i]的收益。因此为了收益最大化，我们列出如下的转移方程：

dp[i][0]=max{dp[i−1][0] , dp[i−1][1]+prices[i]}

再来考虑 dp[i][1]，按照同样的方式考虑转移状态，那么可能的转移状态为前一天已经持有一支股票，即dp[i−1][1]，或者前一天结束时还没有股票，即 dp[i−1][0]，这时候我们要将其买入，则要减去prices[i]的价格。可以列出如下的转移方程：

dp[i][1]=max{dp[i−1][1] , dp[i−1][0]−prices[i]}

1. **dp数组如何初始化**

对于初始状态，根据状态定义我们可以知道第0天交易结束的时候：

dp[0][0]=0，dp[0][1]= − prices[0]

1. **确定遍历顺序**

因此，我们只要从前往后依次计算状态即可。

由于全部交易结束后，不持有股票的收益一定高于持有股票的收益，因此这时候dp[n−1][0]的数值必然是大于dp[n−1][1]的，所以最后的答案即为dp[n−1][0]。

1. **举例推导dp数组**

|  |  |  |  |
| --- | --- | --- | --- |
| 输入  [7,1,5,3,6,4] | i | dp[i][0] | dp[i][1] |
| 0 | 0 | -7 |
| 1 | 0 | -1 |
| 2 | 4 | -1 |
| 3 | 4 | 1 |
| 4 | 7 | 1 |
| 5 | 7 | 3 |

1. **求最优解**

声明两个数组in[n],out[n]，两数组都初始化为0，且分别存储第i天是否有买入操作和第i天是否有卖出操作，若值为1则代表有操作，0代表没有操作。

**out[n]:**

根据dp[i][0]=max{dp[i−1][0] , dp[i−1][1]+prices[i]}可知有两种情况，分别是

①第i-1天就没有股票

代表第i天没有卖出股票，则out[i]=0；

②第i-1天有股票，第i天卖出股票。

代表第i天卖出股票了，则out[i]=1；

因为要选择最优解，所以out[i-1]=0；

**in[n]:**

根据dp[i][1]=max{dp[i−1][1] , dp[i−1][0]−prices[i]}可知有两种情况，分别是

①第i-1天就有股票

代表第i天没有买入股票，则in[i]=0；

②第i-1天没有股票，第i天买入股票。

代表第i天买入股票了，则in[i]=1；

因为要选择最优解，所以in[i-1]=0；

如果dp[0][1] = dp[1][1],则代表手中持有股票是在第一天购入，则in[0]=1，in[1]=0

**注意：由于最后一天一定是卖出股票，所以遍历in[n]时只用遍历到n-1。**

**以上面的例子可以推倒出in[n]和out[n]数组：**

|  |  |  |
| --- | --- | --- |
|  | out[n] | in[n] |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 0 | 1 |
| 4 | 1 | 0 |
| 5 | 0 | 1 |

**由数组可得到最优解:**

第2天买入，第3天卖出，第4天买入，第5天卖出.

1. **代码、结果截图**
2. # include<iostream>
3. **using** **namespace** std;
5. **void** myFrame();
7. **class** Solution {
8. **public**:
10. **int** maxProfit(**int** \*prices,**int** in[],**int** out[],**int** n){
11. **int** dp[n][2],i,j,flag=1;
12. dp[0][0] = 0, dp[0][1] = -prices[0];
13. **for** (**int** i = 1; i < n; ++i) {
14. **if**(dp[i-1][0] > dp[i-1][1]+prices[i]){
15. dp[i][0] = dp[i-1][0];
16. out[i] = 0;
17. }
18. **else** **if**(dp[i-1][0] < dp[i-1][1]+prices[i]){
19. dp[i][0] = dp[i-1][1]+prices[i];
20. out[i-1] = 0;
21. out[i] = 1;
22. }
23. **if**(dp[i - 1][1] > dp[i - 1][0] - prices[i]){
24. dp[i][1] = dp[i - 1][1];
25. in[i]=0;
26. }
27. **else** **if**(dp[i - 1][1] < dp[i - 1][0] - prices[i]){
28. dp[i][1] = dp[i - 1][0] - prices[i];
29. in[i-1] = 0;
30. in[i]=1;
31. }
32. }
33. **if**(dp[0][1]==dp[1][1]) in[0] = 1;  //遍历输出最优解
34. **if**(dp[n - 1][0] == 0) cout<<"没有买入和卖出"<<endl;
35. **else**{
36. **for**(i = 0;i<n-1;i++)
37. **if**(in[i] == 1)
38. cout<<"第"<<i+1<<"天买入    ";
39. cout<<endl;
40. **for**(i = 0;i<n;i++)
41. **if**(out[i] == 1)
42. cout<<"第"<<i+1<<"天卖出    ";
43. cout<<endl;
44. }
45. **return** dp[n - 1][0];
46. }
47. };
49. **int** main(){
50. **while**(**true**){
51. myFrame();
52. **int** n;
53. cout<<"请输入股市一共几天：";
54. cin>>n;
55. **int** prices[n],in[n]={0},out[n]={0};
56. cout<<"请依次输入股票的价格：";
57. **for**(**int** i=0;i<n;i++)
58. cin>>prices[i];
59. Solution s;
60. **int** max;
61. max=s.maxProfit(prices,in,out,n);
62. cout<<"您在该股市最大收益为："<<max<<endl;
63. }
64. }
66. **void** myFrame(){
67. cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;
68. cout<<"\*\*                                    \*\*"<<endl;
69. cout<<"\*\*                                    \*\*"<<endl;
70. cout<<"\*\*                                    \*\*"<<endl;
71. cout<<"\*\*           欢迎来到模拟股市!         \*\*"<<endl;
72. cout<<"\*\*                                    \*\*"<<endl;
73. cout<<"\*\*                                    \*\*"<<endl;
74. cout<<"\*\*                                    \*\*"<<endl;
75. cout<<"\*\*                                    \*\*"<<endl;
76. cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;
77. cout<<"                1.打开模拟市.            "<<endl;
78. cout<<"                2.退出程序.              "<<endl;
79. **int** choice;
80. cout<<"请输入您的选择：";
81. cin>>choice;
82. **if**(choice==2){
83. exit(0);
84. }
85. }

