

华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析

年级：21 级

上机实践成绩：

指导教师：金澈清

姓名：杨茜雅

上机实践名称：贪心算法

学号：

上机实践日期：

10215501435

上机实践编号：No.9

一、目的

1. 熟悉动态规划的基本思想和实现

二、内容与设计思想

1. 完成 OJ 上的两道题目；
2. 分析贪心的过程及答案的正确性；

三、使用环境

推荐使用 C/C++集成编译环境。

四、实验过程

1. 编写相关实现代码；
2. 解释题目中贪心算法的思路，并论证其正确性；
3. 比较两道题目的相同点和不同点；

五、总结

相同点：都是贪心算法，求最优的利润获得。

不同点：一个限制了买卖次数，一个没有。那么针对限制了买卖次数的问题而言就是把 **profit** 在每次循环做一个比较或者更新，更新到最后就是最优解。对于没有限制买卖次数的问题而言则是在把得出最终交易结果的过程分割成很多小过程，通过只加正的 **diprice[]** 保证每个小过程都挣钱，并且没有遗漏任何挣钱的小过程，由此得出最大利润。

买卖问题 1（限定次数）

```
9_1.cpp > main()
1  #include<bits/stdc++.h>
2  #include<algorithm>
3  using namespace std;
4  int main()
5  {
6      int day;
7      cin>>day;
8      int price[100]={0};
9      for(int i=1;i<=day;i++)
10     {
11         cin>>price[i];
12     }
13     int profit=0;
14     int low=price[1];
15     for(int j=1;j<=day;j++)
16     {
17         low=min(low,price[j]);
18         profit=max(profit,price[j]-low);
19     }
20     cout<<profit;
21     return 0;
22 }
```

思路：循环求到第 j 天为止的最大利润，即先找到到第 j 天为止，最低的价格作为买入价，再比较是在第 j 天卖出利润大，还是在前 j 天的某一天卖出利润大，如果是前者的话就对 $profit$ 做一个更新，所以 $Profit$ 的定义是到第 j 天为止，最大的利润，因此等循环遍历完，就是所有天数里，最大的利润。

买卖问题 2（不限定次数）

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int Maxprice=0;
    int day;
    cin>>day;
    int price[100]={0};
    for(int i=1;i<=day;i++)
    {
        cin>>price[i];
    }
    int diprice[100]={0};
    for(int i=2;i<=day;i++)
    {
        diprice[i]=price[i]-price[i-1];
    }
    diprice[1]=0;
    for(int h=1;h<=day;h++)
    {
        if(diprice[h]>0)
            Maxprice+=diprice[h];
        else
            Maxprice+=0;
    }
    cout<<Maxprice;
    return 0;
}
```

思路：因为不限定次数，所以只要买入和卖出的行为会产生顺差即可，所以定义 $diprice[i]$ 数组记录第 $i-1$ 天买入，第 i 天卖出的差价， $diprice[1]$ 定义为 0，因为不存在第 0 天买入第一天卖出的行为。然后在这些差价，也就是 $diprice[]$ 中找出正的数加起来，即为在不限定买卖次数的情况下的最大收益。相当与把 $profit$ 分割成很多隔天进行的交易，在这些交易里挑挣钱的做，把挣钱的小交易都跳出来了，那么 $profit$ 一定是最大的。

$Diprice[]$ 大于 0 的情况可看成是 $t+1$ 交易制度下，所有可以产生顺差的交易，如果把这些都那加起来，则证明在这些天里能挣钱的行为都做了，那么结果一定是最大利润。