**《算法设计与分析》实验报告**

# 实验二 动态规划

|  |  |  |  |
| --- | --- | --- | --- |
| **报告书** | | | |
| 姓名 | 吴宇敖 | 指导教师 |  |
| 学号 | 176001752 | 日 期 | 2020.4.5 |
| 班级 | 17计算机软件一班 |  |  |
| **实验内容** | | | |
| **[要求先在OJ上提交通过]**   1. 小K是个苦命的孩子，他的师傅为了多赚钱，以减肥为理由，让他去采药，并说不完成不能吃饭。野地里有许多不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。要求在规定的时间t里，采到的草药的总价值最大。   自从小K成功的完成了前面的采药任务后，师傅还觉得小K可以更廋点，省点吃饭的成本，于是给小K换了一个地方：每种草药可以无限制地疯狂采摘。[OJ]   1. 大家都知道提莫队长喜欢种蘑菇。有一天提莫正走在回约德尔国的路上，忽然看到路上长了很神奇的蘑菇，蘑菇会不断从某处长出来，但是如果不快点（1秒内）采走的话会消失。酷爱蘑菇的提莫马上去采蘑菇。说来小提莫的人品实在是太好了，这蘑菇别处都没有，就会长在TA身旁的10米范围内。蘑菇如果不马上采走就会坏掉，所以提莫队长马上卸下身上的背包去接。但由于小路两边都不能站人，所以他只能在小路上踩。虽然小莫队长也算约德尔国的短跑健将，但是由于手脚太短，小莫每秒钟还是只能移动1米并且只能踩到1米范围内的蘑菇。   为了使问题简化，可以将小路看作从0-10的一维坐标系，开始时提莫站在5的位置，在第一秒，他只能踩到4,5,6这三个位置中其中一个位置上的蘑菇。问提莫最多能采到多少蘑菇？[OJ]  IMG_256  [Hint](1)数组足够大；(2)while(~scanf("%d", &m)) | | | |
| **实验目的** | | | |
| 1）详细了解动态规划的步骤 | | | |
| **实验过程和步骤** | | | |
| **以下仅供参考，请在报告中删除**  **【实验题目】**  小K是个苦命的孩子，他的师傅为了多赚钱，以减肥为理由，让他去采药，并说不完成不能吃饭。野地里有许多不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。要求在规定的时间t里，采到的草药的总价值最大。  自从小K成功的完成了前面的采药任务后，师傅还觉得小K可以更瘦点，省点吃饭的成本，于是给小K换了一个地方：每种草药可以无限制地疯狂采摘。  输入要求：  第一行有2个整数T(1≤T≤100000)和M(1≤M≤100)，一个空格隔开，T代表总共能够用来采药的时间，M代表山洞里的草药的数目。  接下来的M行每行包括两个在1到100之间包括1和100）的整数，分别表示采摘某株草药的时间和这株草药的价值。  输出要求：  1个整数，表示在规定的时间内可以采到的草药的最大总价值。行末有换行。  **【程序代码及注释】**  #include<iostream>  using namespace std;  int main()  {  int T,M;//一共可以采药的时间和药材的数目  cin>>T>>M;  int time[101];//每个药材需要的时间  int price[101];//每个药材的价钱  for(int i=1;i<=M;i++)//初始化  cin>>time[i]>>price[i];  int res[100001]={0};  for(int j=1;j<=T;j++)  {  res[j]=res[j-1];  for(int i=1;i<=M;i++)  if(j>=time[i])//当时间大于等于采药时间，比较不采这一株药的价值与采了这一株药的价值  res[j]=max(res[j],res[j-time[i]]+price[i]);  }  cout<<res[T]<<endl;  //system("pause");  return 0;  }  **【实验结果及遇到的问题】**    **【算法分析】**  将时间从小到大进行增加，每次都不断的计算，该时间时的最大采药价值。第n的时间进行采药价值的计算时，先将每个要进行循环，当药材的时间小于采药时间时，将采了这一株药和不采这一株药的价值进行比较，记录两者的最大值。  **【实验题目】**    输入要求：  输入数据有多组。多组数据的第一行以正整数n(0<n<100000),表示有n个蘑菇会出现在这条路上。在接下来的n行中，每行有两个整数x,T(0<T<100000),表示在第T秒有一个蘑菇出现在x点上。同一秒钟在同一个点上可能出现多个蘑菇。n=0时输入结束。  输出要求：  每一组输入数据对应一行输出。输出一个整数m，表示提莫队长最多可能踩到m个蘑菇。提示：本题的输入数据量比较大，建议用scanf读入，用cin可能会超时  **【程序代码及注释】**  #include<iostream>  using namespace std;  int a[100001][11]={0};  int main()  {  int x,y,n,m,t;  while(scanf("%d",&n)!=EOF){  //cin>>n;  if(n==0)//输入数据以0结尾  break;  t=0;//初始化时间  for(int i=0;i<100001;i++)//初始化数组  for(int j=0;j<11;j++)  a[i][j]=0;  for(int i=0;i<n;i++){//赋值数据  scanf("%d%d",&x,&y);  a[y][x]++;  if(y>t)//记录最长时间  t=y;  }  for(int i=t-1;i>=0;i--)//从倒数第二秒开始递推  for(int j=0;j<11;j++){  m=a[i+1][j];//选出同一时间三个点中最大的那个  if(j>0&&a[i+1][j-1]>m)  m=a[i+1][j-1];  if(j<10&&a[i+1][j+1]>m)  m=a[i+1][j+1];  a[i][j]+=m;//将最大的那个值加到当前的值  }  cout<<a[0][5]<<endl;//起始点为5  }  //system("pause");  return 0;  }  **【实验结果及遇到的问题】**    遇到问题：  1.一开始没了解题目意思，没有进行不断输入，只能适应一组数据。  2.在oj上提交时不断的出现超时的错误，将while改为while(scanf("%d",&n)!=EOF)之后就解决了这个问题。  **【算法分析】**  对于数据输入时，使用scanf进行输入，将数据使用二维数组进行保存，第一项为时间，第二项为位置，数值为出现的个数，接下来倒着进行递推，先计算倒数第二秒，倒数第二秒每个位置的最大值，应该为那个位置的值加上第n秒时，那个位置附近的三个点的最大值，以此递推，直到第0秒，输出第0秒时，第五个位置的值。 | | | |