

# 《数据结构与算法》实验报告

年级、专业、班级		2021 级计算机学院计算机科学与技术 卓越 2 班		姓名	文红兵
实验题目	动态规划算法实践				
实验时间	2022.11.24	实验地点	竹园四栋		
实验成绩		实验性质	<input type="checkbox"/> 验证性 <input checked="" type="checkbox"/> 设计性 <input type="checkbox"/> 综合性		
<p>教师评价：</p> <div><input type="checkbox"/>算法/实验过程正确；    <input type="checkbox"/>源程序/实验内容提交    <input type="checkbox"/>程序结构/实验步骤合理；</div> <div><input type="checkbox"/>实验结果正确；    <input type="checkbox"/>语法、语义正确；    <input type="checkbox"/>报告规范；</div> <p>其他：</p> <p>评价教师签名：</p>					
<p>实验目的</p> <div>1. 掌握动态规划算法的基本原理</div> <div>2. 训练使用动态规划算法设计技术，通过编程解决不同难度问题的实践能力</div>					
<p>二、实验项目内容</p> <div><div>实验课题 1:</div><p>题目内容：</p><div>7-1 真实的背包故事</div><p>解题思路：</p><div><p><math>dp[i][j]</math>:代表前 <math>i</math> 件物品背包容量为 <math>j</math> 的最大价值</p><p>状态转换方程: <math>dp[i][j] = \min\{dp[i-1][j], dp[i-1][j-w[i]]+v[i]\}</math></p></div><p>源代码：</p><div>最少数量：通过标记数组回溯搜索</div></div>					

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  int main(){
5      // 输入
6      int N,W;
7      cin>>N>>W;
8      int v[N],w[N];
9      for(int i=0;i<N;++i) cin>>v[i]>>w[i];
10
11     // 滚动数组优化 dp[i][j]:代表前i件物品背包容量为j的最大价值
12     int dp[W+1];
13     int s[N][W+1]; // 回溯搜索个数
14
15     // 初始化
16     memset(s,0,sizeof(s));
17     dp[0] = 0;
18     for(int i=1;i<=W;++i) dp[i] = INT_MIN;
19
20     //填表
21     for(int i=0;i<N;++i){
22         for(int j=W;j>=w[i];--j){
23             int temp = dp[j-w[i]]+v[i];
24             if(dp[j]<=temp){
25                 dp[j] = temp;
26                 s[i][j] = 1; // 记录当前
27             }
28         }
29     }

```

```

31     // 回溯搜索路径
32     vector<int> path;
33     int i=N-1;
34     int j=W;
35     while(i>0 && j>0){
36         if(s[i][j] == 1){
37             path.push_back(i);
38             j = j-w[i];
39         }
40         i--;
41     }
42
43     cout<<dp[W]<<" "<<path.size()<<endl;
44     system("pause");
45     return 0;
46 }

```

时间与空间复杂度分析:

时间复杂度是  $O(n^2)$

空间复杂度是  $O(n^2)$

## 实验课题 2:

题目内容:

7-2 动态规划

解题思路:

string dp[maxN][4]: dp[i][j]表示下标 i 前的字符串分割成 j 段的反转后的最小字符串

状态转移方程:  $dp[i][j] = \min(dp[i][j], dp[k][j-1] + temp);$  // k

源代码:

### 时间与空间复杂度分析:

空间复杂度为  $O(n)$

### (3) 思考题

(1) 每件商品的价值相同

(2) 每件商品的重量相同

(3) 每件商品的性价比相同 (性价比 = 价值/重量)

## 没有更好的算法解决

如背包容量:5

商品:  $(1, 1), (1, 1), (1, 1), (1, 1), (5, 5)$

(4) 重量越大, 性价比越低

## 没有更好的算法解决

如背包容量:5

商品: (1, 10), (2, 18), (3, 24), (4, 28), (5, 30)

(5) 重量越大, 价值越小

贪心算法: 对重量排序, 每次选择重量最小的, 直到装不下

时间复杂度:  $O(n \log n)$  空间复杂度:  $O(1)$