广东金融学院实验报告

课程名称：算法分析与设计

装订线

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验编号  及实验名称 | 算法分析与设计实验8 | | | 系 别 | 计算机科学与技术 |
| 姓 名 | 林旋华 | 学 号 | 181543306 | 班 级 | 18计科3 |
| 实验地点 | 电教503 | 实验日期 | 2020.11.12 | 实验时数 | 2 |
| 指导教师 | 郭艺辉 | 同组其他成员 | 无 | 成 绩 |  |
| 1. 实验目的及要求   1) 掌握动态规划、贪心算法的基本思想以及基本原理。  2) 掌握使用动态规划、贪心算法求解问题的一般特征以及步骤。  3) 掌握动态规划、贪心算法设计方法以及复杂性分析方法。  4) 掌握动态规划解0-1背包以及贪心法解活动安排问题算法设计思想、设计过程以及程序实现。 | | | | | |
| 1. 实验环境及相关情况（包含使用软件、实验设备、主要仪器及材料等）   1) 操作系统：Windows操作系统  2) 开发工具：Eclipse、JDK  3) 开发语言：Java | | | | | |
| 1. 实验内容及步骤（包含简要的实验步骤流程）   1、0-1背包的问题提出是，有*n*个物品，其中物品*i*的重量是，价值为，有一容量为*C*的背包，要求选择若干物品装入背包，使装入背包的物品总价值达到最大。此问题的形式化描述是：给定，要求找出*n*元0-1向量, , ，使得目标函数达到最大，并且要满足约束条件。设。  1）采用动态规划算法解决该0-1背包问题，将算法编程实现并截屏实验结果。  2）分析算法复杂性。  2、活动安排的问题提出是，设在活动安排中，每个活动*i*都有一个开始时间和一个结束时间，且，即每个活动在一个半闭区间占用资源，如表1和图1所示。求最优活动安排方案，使得安排的活动个数达到最多。采用贪心算法求解该活动安排问题。   1. 活动安排问题若选择具有最早完成时间的相容活动作为贪心选择可以得到问题的整体最优解。举出3种不能保证产生最优解的贪心选择方案。 2. 将算法编程实现，并分析算法的时间复杂性。   表**1** 活动安排问题   |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 活动***i*** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | | 开始时间***s*[*i*]** | 2 | 1 | 6 | 7 | 11 | 6 | 4 | 14 | 10 | 18 | 16 | 7 | | 结束时间***f*[*i*]** | 4 | 5 | 9 | 10 | 14 | 15 | 16 | 17 | 19 | 22 | 23 | 25 |   图1 活动安排问题示意图 | | | | | |
| 1. 实验结果（包括程序或图表、结论陈述、数据记录及分析等，可附页）   1、0-1背包的问题  1）**public** **class** knapscak {  **public** **static** **void** knapsack(**int** []v,**int** []w,**int** c, **int** [][]m) {  **int** n = v.length - 1;  **int** jMax = Math.*min*(w[n] - 1, c);  **for**(**int** j = 0;j <= jMax;j++)  m[n][j]=0;  **for**(**int** j = w[n];j <= c;j++)  m[n][j]=v[n];    **for**(**int** i = n-1;i > 1;i--) {  jMax = Math.*min*(w[i], c);  **for**(**int** j = 0;j <= jMax;j++)  m[i][j]=m[i+1][j];  **for**(**int** j = w[i];j <= c;j++){  m[i][j]=Math.*max*(m[i+1][j],m[i+1][j-w[i]]+v[i]);  }  m[1][c] = m[2][c];  **if**(c >= w[1])  m[1][c] = Math.*max*(m[1][c], m[2][c-w[1]]+v[1]);  }  }  **public** **static** **void** traceback(**int** [][]m,**int** []w,**int** c,**int** []x)  {  **int** n = w.length - 1;  **for**(**int** i = 1;i < n;i++) {  **if**(m[i][c] == m[i+1][c])x[i] = 0;  **else** {  x[i] = 1;  c -= w[i];  }  }  x[n] = (m[n][c]>0)? 1 : 0;  }  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  **int**[] v = **new** **int**[]{0,20,6,8,15,18};  **int**[] w = **new** **int**[]{0,5,4,8,6,9};  **int** c = 18;  **int** n = v.length;  **int**[][] m = **new** **int**[n+1][c+1];  **int**[] x = **new** **int**[n];  *knapsack*(v,w,c,m);  *traceback*(m,w,c,x);  System.***out***.println("最大价值为:"+m[1][c]);  System.***out***.print("物品重量为:");  **for**(**int** i = 0;i < x.length; i++){  **if**(x[i] == 1){  System.***out***.print(w[i]+" ");  }  }  }  }       1. 算法复杂度：   ①时间复杂度：O(n)  ②空间复杂度：O(n)   1. 活动安排的问题   1）3种方案：  ①每次从剩余活动中选择具有最早开始时间的活动，但是一旦这个活动同时具有最晚结束时间，就得不到问题的最优解。  ②每次从剩余活动中选择具有最短运行时间的活动，但是一旦这个活动同时具有最晚开始时间，就得不到问题的最优解。  ③每次从剩余活动中选择具有最早结束时间的活动，如果每一步都按照这种选择来安排活动，那么显然总可以为剩余未安排的活动留下尽可能多的时间，即使得剩余的可安排时间段极大化，以便安排尽可能多的相容活动。  2）**public** **class** greedySelector {  **public** **static** **int** greedySelector(**int**[] s,**int**[] f,**boolean** a[])  {  **int** n = s.length - 1;  a[1] = **true**;  **int** j = 1;  **int** count = 1;  **for**(**int** i = 2;i <= n; i++)  {  **if**(s[i] >= f[j])  {  a[i] = **true**;  j = i;  count++;  }  **else** a[i] = **false**;  }  **return** count;  }  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  **int**[] s = **new** **int**[]{2,1,6,7,11,6,4,14,10,18,16,7};  **int**[] f = **new** **int**[]{4,5,9,10,14,15,16,17,19,22,23,25};  **boolean**[] a = **new** **boolean**[s.length];  **int** cnt = *greedySelector*(s,f,a);  System.***out***.println("活动最多有" + cnt + "个");  }  }     * 此算法只有一个循环，每个活动只会遍历一次，所以其时间复杂度为O(n)。 | | | | | |
| 1. 实验总结（包括心得体会、问题回答及实验改进意见，可附页）   通过这次实验，对动态规划法求01背包问题和用贪心算法求活动安排问题有更深的理解。其实无非就是抓住书上的递推公式进行写，动态规划依赖于上一个或者上一行的解，就 是在输出子序列的时候有问题。自己对动态规划、贪心、回溯法、分支限界法 的原理不是非常的理解，花了很多时间看了课本上的相关内容。同时课本所提供的代码也是不能直接翻译过来用，当懂得算法的基本原理后，会发现数组下标会出错，所以在参考课本所提供的代码的同时，必须结合算法的实际情况对代码中的相关变量进行修改，这样才能充分利用课本所提供的代码完成本次实 验。通过本次试验，自己基本上掌握上述算法原理，达到实验的目的。 | | | | | |
| 六、教师评语  1、完成所有规定的实验内容，实验步骤正确，结果正确；  2、完成绝大部分规定的实验内容，实验步骤正确，结果正确；  3、完成大部分规定的实验内容，实验步骤正确，结果正确；  4、基本完成规定的实验内容，实验步骤基本正确，所完成的结果基本正确；  5、未能很好地完成规定的实验内容或实验步骤不正确或结果不正确。  6、其它：  评定等级：优秀 良好 中等 及格 不及格  教师签名：郭艺辉 | | | | | |