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 | 实验4 | | | 系别 | 互联网金融与信息工程学院 |
| 姓名 | 谢潮锋 | 学号 | 17B543157 | 班级 | 1715431 |
| 实验地点 | 实验楼503 | 实验日期 | 10月31日 | 实验时数 |  |
| 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) 掌握动态规划算法的基本思想以及基本原理。  2) 掌握使用动态规划算法求解问题的一般特征以及步骤。  3) 掌握动态规划算法设计方法以及复杂性分析方法。   1. 掌握动态规划法解最优二叉搜索树与电路布线问题算法设计思想、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） 2. 最优二叉搜索树问题的问题提出是，设*S*={*x*1, *x*2, ..., *x*n}是一个由*n*个关键字组成的线性有序集，(*a*0, *b*1, *a*1, ..., *bn*, *an*) 为集合*S*的存取概率分布，表示有序集*S*的二叉搜索树利用二叉树的节点存储有序集中的元素。在二叉搜索树中搜索一个元素*x*。在二叉搜索树的内部结点中找到*x*的概率为*bj*；在二叉搜索树的叶结点中确定*x*的概率为*ai*。最优二叉搜索树问题要求找出搜索成本最低的二叉搜索树。设*n*=4，*b*(1:4)=(3, 3, 1, 1)，*a*(1:5)=(2, 3, 1, 1, 1)。 3. 写出最优二叉搜索树问题最优值递归定义。 4. 采用动态规划算法求该最优二叉搜索树，将算法编程实现。 5. 分析算法的时间复杂性。 6. 电路布线的问题提出是，在一块电路板的上、下两端分别有*n*个接线柱，用导线将上端接线柱与下端接线柱相连，其中是的一个排列。要求将这*n*条导线分布到若干绝缘层上，当且仅当两条导线之间无交叉才可以设在同一层。电路布线问题要求确定一个能够布设在同一层的导线集的最大不相交子集。设。 7. 写出电路布线问题最优值递归定义。 8. 采用动态规划算法求解该电路布线问题，将算法编程实现。 9. 分析算法的时间复杂性。 | | | | | |
| 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)    (2)**public** **class** Test {    **public** **static** **void** main(String[] args) {  **int** []a = {2, 3, 1, 1, 1};  **int** []b = {0, 3, 3, 1, 1};  **int** n = a.length - 1;  **int**[][] m = **new** **int**[n+2][n+2];  **int**[][] s = **new** **int**[n+2][n+2];  **int**[][] w = **new** **int**[n+2][n+2];  BinarySearchTest bi = **new** BinarySearchTest();  bi.optimalBinarySearchTree(a, b, m, s, w);  System.***out***.println("最小平均路长为：" + m[1][n]);  }    **public** **void** optimalBinarySearchTree(**int**[] a,**int**[] b,**int**[][] m,**int**[][] s,**int**[][] w){  **int** n = a.length - 1;  **for**(**int** i=0; i<=n; i++){  w[i+1][i] = a[i];  m[i+1][i] = 0;  }  **for**(**int** r=0; r<n; r++){  **for**(**int** i=1; i <= n-r; i++){  **int** j = i+r;    w[i][j] = w[i][j-1] + a[j]+b[j];  m[i][j] = m[i+1][j];  s[i][j] = i;  **for**(**int** k = i+1; k <= j; k++){  **int** temp = m[i][k-1] + m[k+1][j];  **if**(temp < m[i][j]){  m[i][j] = temp;  s[i][j] = k;  }  }  m[i][j] += w[i][j];  }  }  }  }  (3)算法复杂性：**S(n)=O(n3)**  **1.png**  2.  (1)  当i=1时    当i>1时    (2)  **public** **class** Test2 {  **public** **int**[] c;  **public** **int**[][] size;  **public** **int**[] net;  **public** dianlubuxian(**int**[] cc){  **this**.c=cc;  **this**.size=**new** **int**[cc.length][cc.length];  **this**.net=**new** **int**[cc.length];  }  **public** **static** **void** mnset(**int** []c,**int** [][]size)  {  **int** n=c.length-1;  **for**(**int** j=0;j<c[1];j++)  size[1][j]=0;  **for**(**int** j=c[1];j<=n;j++)  size[1][j]=1;  **for**(**int** i=2;i<n;i++) {  **for**(**int** j=0;j<=c[i];j++)  size[i][j]=size[i-1][j];  **for**(**int** j=c[i];j<=n;j++)  size[i][j]=Math.*max*(size[i-1][j], size[i-1][c[i]-1]+1);  }  size[n][n]=Math.*max*(size[n-1][n], size[n-1][c[n]-1]+1);  }  **public** **static** **int** traceback(**int** []c,**int** [][]size,**int** []net) {  **int** n=c.length-1;  **int** j=n;  **int** m=0;  **for**(**int** i=n;i>1;i--)  **if**(size[i][j]!= size[i-1][j]) {  net[m++]=i;  j=c[i]-1;}  **if**(j>=c[1])  net[m++]=1;  System.***out***.println("最大不相交连线分别为：");  **for** (**int** t = m - 1; t >= 0; t--) {  System.***out***.println(net[t]+" "+c[net[t]]);  }  **return** m;    }  **public** **static** **void** main(String[] args) {  **int**[] c = {0,6,8,12,2,1,4,5,3,11,7,10,9,13};  dianlubuxian di = **new** dianlubuxian(c);  di.*mnset*(di.c, di.size);  **int** x = di.*traceback*(di.c, di.size, di.net);  System.***out***.println("最大不相交连线数目为:"+x);  }  }  (3)算法复杂性：**T(*n*)=O(*n*2)**  2.png | | | | | |
| 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） | | | | | |
| 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 | | | | | |