**杭州师范大学杭州国际服务工程学院（信息科学与工程学院）**

班级： 学号： 姓名： 标准答案

装 订 线

**2014-2015学年第2学期期末考试**

**《算法分析与设计》试卷（B）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题号 | 一 | 二 | 三 | 四 | 总分 |
| 得分 |  |  |  |  |  |
| 评卷签名 |  |  |  |  |  |

|  |  |
| --- | --- |
| 得分 |  |

# 填空（共20分，每空格2分）

1．算法是一个由 有限的指令集 组成的过程。

2．算法具有五个特点： 输入、输出、有限性、确定性、可行性 。

3．随机算法应用的例子： 串的相等性判断，素数测试 。

4．不相交集数据结构的合并的两种方法是： 按秩合并措施，路径压缩 。

5．基于递归的技术的常用算法有： 归纳法，分治法，动态规划法 。

6．举两个用分治法解决的典型问题： 二分搜索，大整数乘法，最近点对问题，矩阵乘法 。

7．图的最近点对问题是可以在 多项式（或O(n3)） 时间内求解。

8．举出三个NP完全问题：哈密尔顿回路，旅行商问题，划分问题，装箱问题，图的着色问题。

9．分支限界法包含三个基本过程： 穷举所有状态、广度搜索，剪枝 。

10．图的搜索方法分成两类： 深度优先搜索， 广度优先搜索 。

|  |  |
| --- | --- |
| 得分 |  |

　　　［评分标准：每个空格２分，答对一个给１分］

# 选择（共10分，每小题2分。把最恰当的答案题号填入括号内）

1．下面的函数的时间复杂度为O(*n*3)是（ A ）。

A、（3*n*4+log log *n* ）/（5*n*-100） B、(*n*6+*n*)\*(*n*+log *n*2)

C、*n*!+(*n*+2)\**n*  D、29*n*3+2n+*n*\*log *n*

2．插入排序算法的时间复杂度为（ A ）。

A、 O(*n*2 ) B、O(*n*3) C、O(log *n*) D、O(*n*\*log *n*)

3．利用动态规划求解矩阵连乘问题的时间复杂度为（ B ）。

A、 O(*n*2 ) B、O(*n*3) C、O(log *n*) D、O(*n*\*log *n*)

4．下列可以用近似算法解决的NP问题是：( A,C )

A、装箱问题 B、所有点对的最短路径问题

C、背包问题 D、最小生成树问题

5. 单源点最短路径问题，常用的解决算法是（ B ）

A、Prim算法 B、Dijkstra算法

C、Kruskal 算法 D、Flody算法

［评分标准：第４个选对一个得１分。］

|  |  |
| --- | --- |
| 得分 |  |

# 图解题(四选三，每题10分，共**3**0分)

* 1. 利用Prim算法，以V1为起点求出下图的最小生成树，要求给出步骤。

7

5

1

2

v1

v3

v5

v6

4

1

2

3

6

图1

解：（每步２分，以2-6步中，相邻两个步骤之间变化正确为准）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 步骤 | VT | ～VT | VT与～VT之间的  最短边 | 最短边的权 | 树权 |
| 1 | v1 | v2,v3,v4,v5,v6 | (v1,v2) | 1 | 1 |
| 2 | v1,v2 | v3,v4,v5,v6 | (v2,v3) | 2 | 3 |
| 3 | v1,v2,v3 | v4,v5,v6 | (v3,v5) | 1 | 4 |
| 4 | v1,v2,v3,v5 | v4,v6 | (v5,v4) | 3 | 7 |
| 5 | v1,v2,v3,v5,v4 | v6 | (v4,v6) | 2 | 9 |
| 6 | v1,v2,v3,v5,v4,v6 | Φ | ----- | ----- | ------ |

最终得到的MST为：

v4

v2

2

1

v6

2

3

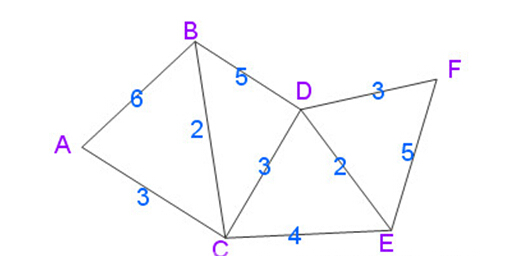
v1

1

v3

v5

* 1. 利用Dijkstra算法，求出下图以A为单源点出发的所有最短距离，要求给出计算步骤。



解：【评分标准：每步２分，以2-6步相邻两个步骤之间变化正确为准】

算法执行步骤如下表：

|  |  |  |
| --- | --- | --- |
| **步骤** | **S集合中** | **U集合中** |
| 1 | 选入A，此时S=<A>  此时最短路径A→A=0  以A为中间点，从A开始找 | U=<B、C、D、E、F>  A→B=6，A→C=3，  A→其它U中的顶点=∞，  发现A→C=3权值为最短 |
| 2 | 选入C，此时S=<A、C>  此时最短路径A→A=0，A→C=3以C为中间点，  从A→C=3这条最短路径开始找 | U=<B、D、E、F>  A→C→B=5(比上面第一步的A→B=6要短)  此时到D权值更改为A→C→B=5，  A→C→D=6，  A→C→E=7，  A→C→其它U中的顶点=∞，发现A→C→B=5权值为最短 |
| 3 | 选入B，此时S=<A、C、B>  此时最短路径A→A=0，A→C=3，A→C→B=5  以B为中间点  从A→C→B=5这条最短路径开始找 | U=<D、E、F>  A→C→B→D=10(比上面第二步的A→C→D=6要长)  此时到D权值更改为A→C→D=6，  A→C→B→其它U中的顶点=∞，发现A→C→D=6权值为最短 |
| 4 | 选入D，此时S=<A、C、B、D>  此时最短路径A→A=0，A→C=3，A→C→B=5，A→C→D=6  以D为中间点，  从A→C→D=6这条最短路径开始找 | U=<E、F>  A→C→D→E=8(比上面第二步的A→C→E=7要长)此时到E权值更改为A→C→E=7，A→C→D→F=9  发现A→C→E=7权值为最短 |
| 5 | 选入E，此时S=<A、C、B、D、E>  此时最短路径A→A=0，A→C=3，A→C→B=5，A→C→D=6，A→C→E=7，以E为中间点，  从A→C→E=7这条最短路径开始找 | U=<F>  A→C→E→F=12(比上面第四步的A→C→D→F=9要长)此时到F权值更改为A→C→D→F=9  发现A→C→D→F=9权值为最短 |
| 6 | 选入F，此时S=<A、C、B、D、E、F>  此时最短路径A→A=0，A→C=3，  A→C→B=5，  A→C→D=6，  A→C→E=7，A→C→D→F=9 | U集合已空，查找完毕。 |

* 1. 给出下图的有向图从顶点a开始运行深度优先搜索饿结果，给出边的分类。



解：（搜索树６分，边的分类４分）



树边：a-b,b-f,b-e,c-d

前向边：a-f

回边：e-a

横跨边：f-e,d-e

* 1. 设m元钱，n项投资，函数 表示将x元投入第i项项目所产生的效益，i=1,2,…,n.问：如何分配这m元钱，是的投资的总效益最高？

假设分配给第i个项目的钱数是xi，问题描述为：

目标函数 

约束条件 

设表示x万元投给前k个项目的最大效益，k=1,2,…,n，x=1,2,…,m

根据下表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 投资x (万元) | 效益万元 |  |  |  |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 11 | 0 | 2 | 20 |
| 2 | 12 | 5 | 10 | 21 |
| 3 | 13 | 10 | 30 | 22 |
| 4 | 14 | 15 | 32 | 23 |
| 5 | 15 | 20 | 40 | 24 |

填写该表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |

解答：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x |  |  |  |  |  |  |  |  |
| 1 | ***11*** | ***1*** | ***11*** | ***0*** | 11 | 0 | 20 | 1 |
| 2 | 12 | 2 | 12 | 0 | 13 | 1 | 31 | 1 |
| 3 | 13 | 3 | 16 | 2 | 30 | 3 | 33 | 1 |
| 4 | 14 | 4 | 21 | 3 | ***41*** | ***3*** | 50 | 1 |
| 5 | 15 | 5 | 26 | 4 | 43 | 4 | ***61*** | ***1*** |

|  |  |
| --- | --- |
| 得分 |  |

# 问答题（六选四，每题10分，共40分）

1. 应用分支限界法求解以下旅行商问题的实例（10分）：

班级： 学号： 姓名：

装 订 线

v1 v2 v3 v4 v5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ∞ | 17 | 7 | 35 | 18 |
| 9 | ∞ | 5 | 14 | 19 |
| 29 | 24 | ∞ | 30 | 12 |
| 27 | 21 | 5 | ∞ | 48 |
| 15 | 16 | 28 | 18 | ∞ |

v1

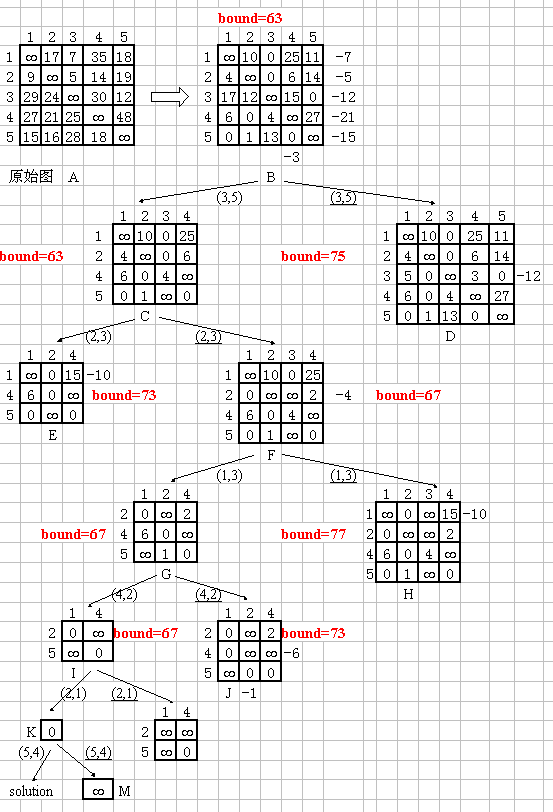
v2

v3

v4

v5

解：（每步各１分）



|  |  |
| --- | --- |
| 得分 |  |

1. 简述求平面中n 个点的集合S的极大点集合的几何扫描算法。（10分）

解：设p1=(x1,y1),p2=(x2,y2)是平面上的两个点，如果x1<=x2 且y1<=y2,则称p2支配p1,记为p1<p2。

设S是平面中的一个点集，点p ∈S是极大点或最大点，如果不存在点　q∈ S ,使得p≠q并且p<q。

算法：把S中的所有点按照它们x坐标的非升序排列，最右点（有最大x值的那点）无疑是个最大点。从右到左扫描这些点，同时确定它是否在y坐标上被先前扫描过的任何点所支配。

输入：平面上n个点的集合S.

输出：S中极大点集合M。

1. 设A为S中按x坐标非生序排列的点的集合。如果两个点有相同的x坐标，则有较大y坐标值的点出现在集合的前面。
2. M={A[1]}
3. Maxy=A［１］的y坐标值。
4. For j=2 to n
5. (x,y)=a[j]
6. if y>maxy then
7. M=M U {A[j]}
8. maxy=y
9. endif
10. endfor

［评分标准：给出算法的每一步，各１分］

1. 用动态规划算法求解最长公共子序列问题。（10分）

|  |  |
| --- | --- |
| 得分 |  |

这2个序列是：

A={xyxxzxyzxy}

B={zxzyyzxxyxxz}

解：•若给定序列X={x1,x2,…,xm}，则另一序列Z={z1,z2,…,zk}，是X的子序列是指存在一个严格递增下标序列{i1,i2,…,ik}使得对于所有j=1,2,…,k 有：zj=xij。

• 给定2个序列X和Y，当另一序列Z既是X的子序列又是Y的子序列时，称Z是序列X和Y的**公共子序列**。

设序列X={x1,x2,…,xm}和Y={y1,y2,…,yn}的最长公共子序列为Z={z1,z2,…,zk} ，则

(1)若xm=yn，则zk=xm=yn，且zk-1是xm-1和yn-1的最长公共子序列。

(2)若xm≠yn且zk≠xm，则Z是xm-1和Y的最长公共子序列。

(3)若xm≠yn且zk≠yn，则Z是X和yn-1的最长公共子序列。

由最长公共子序列问题的最优子结构性质建立子问题最优值的递归关系。用c[i][j]记录序列和的最长公共子序列的长度。其中， Xi={x1,x2,…,xi}；Yj={y1,y2,…,yj}。当i=0或j=0时，空序列是Xi和Yj的最长公共子序列。故此时C[i][j]=0。其它情况下，由最优子结构性质可建立递归关系如下：



void LCSLength(char x[], char y[],int m，int n)

{ int L[m][n],i，j;

for (i = 0; i <= m; i++) L[i][0] = 0;

for (i = 0; i <= n; i++) L[0][i] = 0;

for (i = 1; i <= m; i++)

for (j = 1; j <= n; j++) {

if (x[i]==y[j])

L[i][j]=L[i-1][j-1]+1;

else if (L[i-1][j]>= L[i][j-1])

L[i][j]= L[i-1][j];

else L[i][j]= L[i][j-1];

return L[m][n];

}

LCS(x,y)={xyxxxz}

［评分标准：给出算法思想或完整代码给１０分］

1. 描述利用回溯法求解8皇后问题。（10分）

|  |  |
| --- | --- |
| 得分 |  |

解：在8×8格的棋盘上放置彼此不受攻击的8个皇后。按照国际象棋的规则，皇后可以攻击与之处在同一行或同一列或同一斜线上的棋子。8后问题等价于在8×8格的棋盘上放置8个皇后，任何2个皇后不放在同一行或同一列或同一斜线上.

•解向量：(x1, x2, … , xn)

•显约束：xi=1,2, … ,n

•隐约束： 1)不同列：xixj 2)不处于同一正、反对角线：|i-j||xi-xj|

bool **Place**(int k)

{ for (int j=1;j<k;j++)

if ((abs(k-j)==abs(x[j]-x[k]))||(x[j]==x[k])) return false;

return true;

}

void **Backtrack**(int t)

{

if (t>n) return;

else

for (int i=1;i<=n;i++) {

x[t]=i;

if (Place(t)) Backtrack(t+1);

}

}

其中一个解：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Q |  |  |  |  |
|  |  |  |  |  | Q |  |  |
|  |  |  |  |  |  |  | Q |
|  | Q |  |  |  |  |  |  |
|  |  |  |  |  |  | Q |  |
| Q |  |  |  |  |  |  |  |
|  |  | Q |  |  |  |  |  |
|  |  |  |  | Q |  |  |  |

［评分标准：给出算法代码给１０分，给出一个解给３分］

1. 求快速排序法的平均复杂度，并给出分析过程（10分）

假设数组A的首元素在排好序后处在n个位置中的任何位置是等可能的，即它处在任何位置的概率是1/n，如果它处在位置i (i=1,2,…n)，那么划分后的两个字问题规模分别是i-1和n-i.考虑到T(0)=0,因此可以得到

尝试法

设:

左边

右边

积分近似

左边为底长1，高的矩形

令

则

右边

=

1. 给出锦标赛算法求第二大元素，并证明最大元素在第一阶段的分组比较中共计进行了次比较。（10分）

利用锦标赛淘汰规则，先选出冠军，再在与冠军比赛中失败的选手中选取最大值，即为第二大元素。

1. 将n个元素两两一组，分成组
2. 每组的两个数比较，较大的数；
3. 剩余较大的数再两两比较，直到找出冠军；
4. 淘汰n-1个元素，每淘汰一次，需要一次比较，共需要n-1次比较
5. 对每个元素建立被自己淘汰的列表
6. 在冠军对应的列表中选择最大值，第二大元素只能出现在与最大元素max直接比较所淘汰的元素中产生。

复杂度：

**命题 2.2** max（元素）在第一阶段的分组比较中共计进行了次比较。

证明：

第1轮：参与比较n个元素，淘汰后剩余个，进入第2轮

第2轮：参与比较个元素，淘汰后剩余个，进入第3轮

……

第k轮：参与比较个元素，淘汰剩余个

假设k轮淘汰后剩余一个，即

1. 若，那么有
2. 若，则

，由

k最小可取d+1，即；

又

；

因此，

复杂度：

其中次比较选出冠军，在个与冠军淘汰元素中选择一个，需要