华中科技大学计算机科学与技术学院 2021~2022 第一学期

“ 算法设计与分析 ”考试试卷（A 卷）

考试方式 闭卷 考试日期 单击或点击此处输入日期。 考试时长 150 分钟

专业班级 _____ 学 号 _____ 姓 名 _____

题号	一	二	三	四	五	六	七	总分	核对人
分值	24	8	12	16	12	14	14	100	
得分									

分 数	
评卷人	

一、选择题（每小题的 A、B、C、D 四个选项中，有一个或多个选项是正确的，请选择正确的选项。本题共 12 小题，每小题 2 分，共 24 分）

解答内容不得超过装订线

1．以下选项中，属于算法五个重要特性的是 ABC 。

A、确定性 B、有穷性 C、可行性 D、能行性

2、假设 $f(n)$和 $g(n)$是渐近正函数，若存在正常数 c_1、c_2 和 n_0，使得对所有的 $n \geq n_0$ 有 $c_1 g(n) \leqslant f(n) \leqslant c_2 g(n)$，则记作 C 。

A、$f(n)=O(g(n))$ B、$g(n)=O(f(n))$ C、$f(n)=\Theta(g(n))$ D、$g(n)=\Theta(f(n))$

3、如果 $d(n)$是 $O(f(n))$， $e(n)$是 $O(g(n))$，那么 $d(n)+e(n)$是 A 。

A、$O(f(n)+g(n))$ B、$O(d(n)+e(n))$ C、$\Theta(f(n)+g(n))$ D、$\Theta(d(n)+e(n))$

4、以下属于稳定排序的算法有 BCD 。

A、快速排序 B、归并排序 C、插入排序 D、冒泡排序

5、用动态规划策略求解，一般要求问题满足 ABC 。

A、重叠子问题性质 B、最优子结构性 C、无后效性 D、贪心选择性

6、以下属于贪心算法的是 ABC 。

A、Bellman-Ford 算法 B、Prim 算法 C、Dijkstra 算法 D、Floyd-Warshall 算法

7、以下可以使用动态规划进行求解的问题是 BD 。

A、最长公共子序列问题 B、活动选择问题

C、单源点最短路径问题 D、所有结点对之间的最短路径问题

8、以下属于启发式搜索算法的是____B____。

A、BFS　　B、LC-检索　　C、LIFO-检索　　D、FIFO-检索

9、设 G 是一个流网络，则定义在 G 上的流 f 应满足____CD____。

A、容量守恒　　B、流量限制　　C、容量限制　　D、流量守恒

10、关于最优二叉检索树，以下描述正确的是____AB____。

A、最优二叉检索树的加权平均检索次数是最少的

B、相比包含相同结点的其它二叉树，最优二叉检索树的高度是最矮的

C、相比包含相同结点的其它二叉树，最优二叉检索树的叶子结点数是最少的

D、最优二叉检索树可以为单分支的二叉树

11、以下可用于求解递归式的方法有____BC____。

A、列表法　　B、递归树法　　C、主方法　　D、代入法

12、任何以比较为基础的排序算法，最坏情况下的时间下界是____B____。

A、$\Omega(\log n)$　　B、$\Omega(n\log n)$　　C、$\Omega(n^2)$　　D、无法不确定

分　数	
评卷人	

二、求下列递归式的渐近紧确界（本题 8 分）

要求：写出计算过程。

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2\sqrt{n}$$

将式子代入递推关系式$T(n) = aT\left(\frac{n}{b}\right) + f(n)$

$a = 4, b = 2, n^{\log_b a} = n^2$

$f(n) = n^2\sqrt{n} = \Omega(n^{log_b a+\epsilon})(\epsilon \le \frac{1}{2})$

是主定理的第三种情况

$T(n) = \Theta(f(n)) = \Theta(n^2\sqrt{n})$

分　数	
评卷人	

三、简答与计算(本题共 2 小题，每小题 6 分，共 12 分)

1、简述活动选择问题求最大兼容活动集合的贪心算法设计思想，并对以下活动集合（s_i、f_i分别是活动的开始时间和结束时间）求出它的一个最大兼容活动集合，要求：写出一定的计算过程。

i	1	2	3	4	5	6	7	8	9	10
s_i	0	1	2	3	3	5	5	6	8	12
f_i	6	4	14	5	9	7	9	10	11	16

算法设计思想：（核心：贪心选择，每次总选择具有最早结束时间的兼容活动加入到集合中）

步骤：

1. 先把题目中给出的所有活动按照结束时间的递增顺序排序
2. 根据贪心思想，迭代选择结束时间最早的兼容活动（结束时间最早（f 值最小），且时间要能兼容，即开始时间要满足约束（s 值要大于当前集合中的最大结束时间，否则不兼容））

伪代码：

1. 使用 sort() 函数或者自行实现的排序函数对输入的活动按照结束时间排序
2. GREEDY-ACTIVITY-SELECTOR(s,f)://注意如下代码中的下标是排序后的下标，需要映射到原来的下标（下标从 1 开始）

```
n=s.length()
A={a1}
k=1;
for m=2 to n
        if s[m]>= f[k]
                A=A∪{am}
                k=m;
return A
```

结果：

最大兼容活动集合的大小为 4，
包含活动序号为{2, 6, 9, 10}，
对应的开始时间和结束时间为 { (1,4) , (5,7) , (8,11) , (12,16) }

2、请画出下面的差分约束系统的约束图。并回答如何利用约束图求一个差分约束系统的可行解或判定该系统没有可行解。

$x_1-x_2 \leqslant 1$

解答内容不得超过装订线

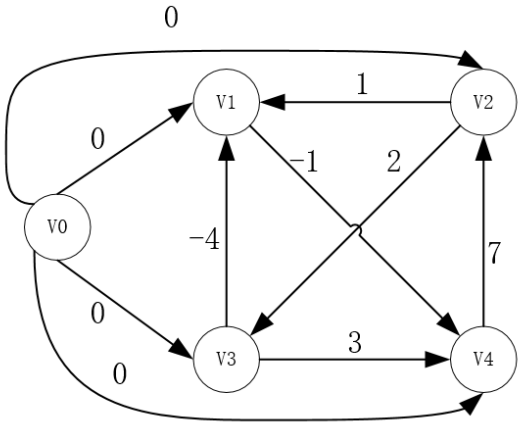
x_1-x_3≤−4

x_2-x_4≤7

x_3-x_2≤2

x_4-x_1≤−1

x_4-x_3≤3



利用约束图求一个差分约束系统的可行解或判定该系统有没有可行解：

方法：

使用 Bellman-Ford 算法求单源最短路径

1. 如果约束图中不包含权重为负值的回路（Bellman-Ford 算法返回 TRUE），则该系统的一个可行解为： 每个 Xi 的值取虚拟源点到 Vi 的最短距离
2. 如果约束图中包含权重为负值的回路(Bellman-Ford 算法将返回 FALSE)，则该系统没有可行解

对于本题数据，可以求得一个可行解为：

x=(-4,0,0,-5);

对于任意常数 d, (d-4,d,d,d-5);也是问题的解

分　数	
评卷人	

四、（本题 16 分）对给定的两个序列 X 和 Y，记 c[i, j]为前缀序列 X_i 和 Y_j 的一个 LCS 的长度：

$$c[i,j]=\begin{cases}0 & \text{如果} i=0 \text{ 或} j=0\\ c[i-1,j-1]+1 & \text{如果 i,j}>0 \text{ 且} x_i=y_j\\ max(c[i,j-1],c[i-1,j]) & \text{如果 i,j}>0 \text{ 且} x_i\neq y_j\end{cases}$$

已知序列 X=< C, B, C, A, B, A, C, B>和 Y=<A, C, B, D, A, B, C>，求 LCS(X, Y)。

j	0	1	2	3	4	5	6	7
i	y_j	A	C	B	D	A	B	C

解答内容不得超过装订线

0	x_i							
		0	0	0	0	0	0	0
1	C		↑	↖	←	←	←	←	↖
		0	0	1	1	1	1	1
2	B		↑	↑	↖	←	←	↖	←
		0	0	1	2	2	2	2
3	C		↑	↖	↑	↑	↑	↑	↖
		0	0	1	2	2	2	2	3
4	A		↖	↑	↑	↑	↖	←	↑
		0	1	1	2	2	3	3	3
5	B		↑	↑	↖	↑	↑	↖	←
		0	1	1	2	2	3	4	4
6	A		↖	↑	↑	↑	↖	↑	↑
		0	1	1	2	2	3	4	4
7	C		↑	↖	↑	↑	↑	↑	↖
		0	1	2	2	2	3	4	5
8	B		↑	↑	↖	←	↑	↖	↑
		0	1	2	3	3	3	4	5

LCS(X,Y)= CBABC 。

分　数	

评卷人	

五、（本题 12 分）数组 A[1..n]中含有 n 个互不相同的整数元素。对 A 中的元素 A[i]（1≤i≤n），若有 A[i]<A[i-1]并且 A[i]<A[i+1]，则称 A[i]为 A 的局部最小元素，即局部最小元素是比其两个相邻元素都小的元素（注：在边界上，即 i=1 或 i=n 时，只需考虑一侧的邻居即可）。例：如果 A={5,3,4,1,2}，那么 A 有二个局部最小元素 3 和 1；而若 A={1,2,3,4,5}，那么 A 就只有一个局部最小值元素 1。

请设计一个时间复杂度为 O(logn)的算法输出 A 中的一个局部最小元素（当有多个局部最小元素时，输出任意一个即可），给出算法的伪代码描述，并证明你的算法关于时间复杂度的结论。

可以使用二分法求解。

1. 首先判断头或者尾是不是局部最小元素，如果是，则可以直接返回
2. 如果头和尾都不符合，则可以用二分法查找到局部最小元素

   初始时，左边界 l=0, 右边界 r=n-1;

   取中间位置 mid, 对应的值为 A[mid]

   则可以根据如下规则查找：

   a. 如果中间位置满足 A[mid]<A[mid-1]&&A[mid]<A[mid+1]，则此时的 A[mid]就是符合条件的局部最小元素，返回这个元素即可，程序结束
   b. 如果 A[mid]>A[mid-1]，则可以判定在[l,mid-1]范围内一定有局部最小元素，修改右边界 r=mid-1
   c. 如果 A[mid]>A[mid+1]，则可以判定在[mid+1,r]范围内一定有局部最小元素

      修改左边界 l=mid+1

时间复杂度分析：

1. 判断特殊情况的复杂度：2*O(1)
2. 二分查找的过程，每次都可缩减到原规模的 1/2，复杂度为 log(n)

因此，总的时间度为 O(logn)

完整代码如下：

```
int findPartMin(vector<int> A,int n){
    if(A[0]<A[1]){//特殊情况 1：头部元素就是局部最小元素
        return A[0];
    }
    else if(A[n-1]<A[n-2]){//特殊情况 2：尾部元素就是局部最小元素
        return A[n-1];
    }
    else{//如果头尾都不是局部最小元素，则二分查找局部最小元素
        int l=0;
        int r=n-1;
        while(l<=r){//二分查找
            int mid=l+(r-l)/2;//中间元素位置
            if(A[mid]<A[mid-1]&&A[mid]<A[mid+1]){
                return A[mid]; //如果中间元素符合则可以直接返回这个中间元素
            }
            else if(A[mid]>A[mid-1]){
                r=mid-1; //此时可以判定在[l,mid-1]范围内一定有局部最小元素
```

```
            }
            else if(A[mid]>A[mid+1]){
                l=mid+1; //此时可以判定在[mid+1,r]范围内一定有局部最小元素
            }
        }
    }
}
```

分　数	
评卷人	

六、(本题 14 分)设有 n 个任务(用编号 1～n 表示),每个任务 j∈[1..n],都有一个权重（记为 W_j)、执行时间（记为 l_j)，其中,$W_j \in [0, 1]$且 $\sum_{j=1}^{n} W_j = 1$；并记其完成时间为 C_j。这里仅考虑任务的串行调度，即一个任务接着一个任务被调度执行，不考虑等待和空闲时间，则 C_j即是串行调度至任务 j 并执行完任务 j 的总时间。

请设计一个调度算法，求出一种执行顺序，使得所有任务按顺序执行完后，$\sum_{j=1}^{n} W_j C_j$最小，并证明算法的正确性。

使用贪心算法，将任务按照$\frac{W}{l}$从大到小进行排序，并按该顺序执行任务。使用反证法证明此时$\Sigma_{j=1}^{n} W_j C_j$最小，若此时$\Sigma_{j=1}^{n} W_j C_j$不是最小，则我们可以找到连续的$i, k (i, k \in \{1, \dots, n\}, i < k)$, 使得其在交换顺序后$\Sigma_{j=1}^{n} W_j C_j$更小，下面证明这是不可能的:

记完成前$i-1$个任务用时为t_1, 完成前k个任务用时为t_2

交换前$T = \Sigma_{j=1}^{n} W_j C_j = \Sigma_{j=1}^{i-1} W_j C_j + (t + l_i) W_i + t_2 W_k + \Sigma_{j=k+1}^{n} W_j C_j$

交换后$T' = \Sigma_{j=1}^{n} W_j C_j = \Sigma_{j=1}^{i-1} W_j C_j + (t + l_k) W_k + t_2 W_i + \Sigma_{j=k+1}^{n} W_j C_j$

$$T' - T = l_k W_i - l_i W_k = l_k l_i \left(\frac{W_i}{l_i} - \frac{W_k}{l_k} \right) > 0$$

即交换后$\Sigma_{j=1}^{n} W_j C_j$变大，由此说明我们算法的正确性

解答内容不得超过装订线

分　数	
评卷人	

七、（本题 14 分）符号乘法问题：定义在符号集 S={a，b，c} 上的一种乘法运算规则如下表所示：

	a	b	c
a	b	b	a
b	c	b	a
c	a	c	c

如，ab=b，ba=c 等。注：该乘法规则不满足结合律和交换律。

请设计一个有效的算法，对给定的 S 上的符号串，如 bbbbac，判定是否可以通过适当加括号的方式，使得其“乘积”等于 a，若可以则返回 TRUE，否则返回 FALSE。如，对 bbbbac，算法返回 TRUE，因为((b(bb))(ba)c=a。

Dp 为字符串第 i-j 个元素组成的元素所能组成的乘积集合

Mul 为题目所给运算规则数组

使用动态规划得到 dp[0][n],并判断 a 是否在其中即可，算法如下：

```
def f(i, j, s):
    if i == j:
        dp[i][j] = set(s[i])
    else:
        for k in range(i, j):
            for c in dp[i][k]:
                for d in dp[k+1][j]:
                    dp[i][j].append(mul[c][d])
    return
```

解答内容不得超过装订线