北 京 交 通 大 学

题号	一	二	三	四	五	六	七	总分
得分								
阅卷人								

（注意：编写程序题目，请在关键代码处添加注释！）

一、单选题（每题 2 分，共 10 分）

1. 某算法的时间复杂度为$O(n^2)$，表明该算法（ C ）。

A.问题规模是n^2

B.执行时间等于n^2

C.执行时间与n^2成正比

D.问题规模与n^2成正比

2.对[15,9,7,8,20,-1,4]进行排序，进行一次后数据的排序变为[9,15,7,8,20,-1,4]，则采用的是（ C ）算法。

A.直接选择排序

B.冒泡排序

C.直接插入排序

D.希尔排序

3. 利用动态规划计算以下矩阵连乘的正确组合是（ A ）。

A1(20*25)、A2(25*5)、A3(5*15)、A4(15*10)、A5(10*20)、A6(20*25)

A. (A1A2)(((A3A4)A5)A6)

B. (A1A2A3)((A4A5)A6)

C. (((A1((A2A3)A4))A5)A6)

D．(A1A2)((A3(A4A5))A6)

4. 能采用贪心算法求最优解的问题，一般具有的重要性质为（ A ）。

A 最优子结构性质与贪心选择性质

B 重叠子问题性质与贪心选择性质

C 最优子结构性质与重叠子问题性质

D 预排序与递归调用

5.考虑以下关于搜索算法的陈述，下列哪项是不正确的（ C ）。

A.搜索算法的效率可能会受到搜索空间大小的影响

B.在搜索算法中，剪枝可以有效地减少不必要的搜索

C.搜索算法不能并行化，因为每个步骤都依赖于上一个步骤的结果

D.搜索算法可以通过启发式方法来提高效率

二、多选题（每题 3 分，共 15 分）

1. 设 n 为算法中的问题规模，通常用（A、C、D ）渐进符号表示算法的执行时间与 n 之间的一种增长关系。

A. Ω

B. Σ

C. O

D. Θ

2. 分治策略所能解决的问题一般具有以下几个特征（A、C、D）。

A. 问题的规模缩小到一定程度就可以容易求解

B. 子问题之间相互重叠

C. 子问题的解可以合并出原问题的解

D. 问题可以分解为若干个规模较小的相同问题

3. 找出 n 个自然数中取 r 个数的组合，例如 n=5，r=3 有 10 种组合，以下关于该问题使用循环和递归实现的说法正确的是（A、B、C）。

A. 当 r=3 时，循环算法设计中可以用三重循环模拟每个组合中 3 个数

B. 当 r=3 时，循环算法的时间复杂度是$O(n^3)$

C. 递归算法实现的时间复杂度是$O(nr)$

D. 递归实现的层次是不能控制的，而循环嵌套的层次只能是固定的

4. 回溯法搜索解空间树时，常用的两种剪枝函数是（A、B）。

A. 约束函数

B. 限界函数

C. 符号函数

D. 三角函数

E. 正切函数

5. 下列叙述不正确的有（A、B、C、E ）。

A. 平衡二叉树的插入节点操作平均时间复杂度是$O(n^2)$

B. 递归的两个基本要素是递归方程和约束函数

C. 折半查找法的查找速度一定比顺序查找法快

D. 采用深度优先搜索或拓扑排序算法可以判断出一个有向图中是否有环(回路)

E. 有 n 个数存放在一维数组 A[1,n]中，在进行顺序查找时，这 n 个数的排列有序或无序其平均查找长度不同

三、编程题

1. 回文日期判定（15 分）

小明习惯用 8 位数字表示一个日期，其中，前 4 位代表年份（年份范围：1000~9999），接下来 2 位代表月份，最后 2 位代表日期。一个日期是回文的，当且仅当表示这个日期的 8 位数字是回文的。现在，小明想知道：在他指定的两个日期之间包含这两个日期本身，有多少个真实存在的日期是回文的。

例如：

- 对于 2010 年 1 月 2 日，用 8 位数字 20100102 表示，它是回文的。
- 对于 2010 年 10 月 2 日，用 8 位数字 20101002 表示，它不是回文的。

每一年中都有 12 个月份：

其中，1、3、5、7、8、10、12 月每个月有 31 天；4、6、9、11 月每个月有 30 天；而对于 2 月，闰年时有 29 天，平年时有 28 天。

一个年份是闰年当且仅当它满足下列两种情况其中的一种：

1.这个年份是 4 的整数倍，但不是 100 的整数倍；

2.这个年份是 400 的整数倍。

例如：2000、2012、2016 是闰年。

输入描述:

输入包括两行，每行包括一个 8 位数字。第一行表示小明指定的起始日期；第二行表示小明指定的终止日期。保证 datei 和都是真实存在的日期，且年份部分一定为 4 位数字，且首位数字不为 0。保证 date1 一定不晚于 date2。

输出描述：

输出一行，包含一个整数，表示在 date1 和 date2 之间，有多少个日期是回文的。

1）设计算法解决这一问题，并给出源程序。（10 分）

2）分析所提出算法的时间复杂度。（5 分）

参考答案

1）算法设计思路：根据日期回文数 8 位数特点，只需要至多枚举 10000 个数字即可（这一问题有多类枚举算法，这是其中之一；源程序完全正确 10 分，如果未考虑闰年扣 2 分；程序不完整酌情扣分；未提供程序，只是给出算法的思路，得 3 分）。

算法源程序

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int b[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};//1 到 12 月的天数
    string s1,s2;
    cin>>s1>>s2;
    int a=0;
    int ans=0;
    for(int i=0;i<10000;i++){//只需枚举到 10000，因为构造的回文串只有 8 位
        string s=to_string(i);
        string s3;
        for(int j=s.size();j<4;j++){
            s3+='0';
        }
        s3+=s;
        string s4=s3;
        reverse(s4.begin(),s4.end());//翻转，构造回文串
        s4+=s3;
        if(s4>=s1&&s4<=s2){
            int c=stoi(s4.substr(0,4));//年
            int d=stoi(s4.substr(4,2));//月
            int e=stoi(s4.substr(6,2));//日
            if(c%400==0||(c%4==0&&c%100!=0)){//如果是闰年，2 月有 29 天
                b[2]=29;
            }
            if(d>=1&&d<=12&&e>=1&&e<=b[d]){//如果枚举的回文是正确的日期，答案+1
                ans++;
            }
        }
    }
    cout<<ans<<endl;
}
```

2)分析算法的时间复杂度（基于所提出的算法,时间复杂度分析正确得 5 分，不正确不得分）

方案 1：对 8 位数字进行枚举的话，复杂度为 O(n)

方案 2：上述程序所对应的算法，复杂度为$O(\sqrt{n})$

2. 元素查找问题（15 分）

n 个不同的整数组成数组 A，按降序排列设计分治算法，判断 A 中是否存在 A[i]=i 的数。

1）给出分治算法代码或者伪代码。 （8 分）

2）分析算法的时间复杂度。 （7 分）

参考答案：

1）采用二分查找方法。a[i]=i 时表示该元素在有序非重复序列 a 中恰好第 i 大。对于序列 a[low..high]，mid=(low+high)/2，若 a[mid]=mid 表示找到该元素；若 a[mid]>mid 说明左区间的所有元素都大于其位置，只能在右区间中查找；若 a[mid]<mid 说明右区间的所有元素都小于其位置，只能在左区间中查找。（程序略，给出完整程序或者伪代码，得 7 分；根据算法代码或伪代码完整性酌情扣分）

2）设 n 个元素数组查找的时间为 T(n)，根据算法得到的时间递推公式为：T

T(1)=1

T(n)=T(n/2)+1

基于递推法或者主定理，算法的时间复杂度为 O(logn)。

3. 最大收益问题（25 分）

n 万元（n 为正整数）投资 k 个项目，$g_i(x)$表示将 x 万元（x 为整数）投入第 i 个项目上的收益。分别用动态规划和搜索算法求解，使得总收益最大的投资方案（每个项目只能投 1 次）。

例：4 万元投资 4 个项目的收益如下表

x	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$
0	0	0	0	0
1	2	12	4	21
2	7	13	11	22

3	12	14	21	23
4	17	15	25	24

1）设计一种动态规划算法，（1）给出动态规划算法的递推方程，并给出该案例的最优解；（2）分析所提出动态规划算法的时间复杂度。（10 分）

2）设计回溯算法，（1）绘制上述案例的状态空间图；（2）描述剪枝函数的设计思路；（3）写出算法代码或伪代码。（15 分）

参考答案：

1）问题的解向量是$(x_1, x_2, \dots, x_k)$，x_i是投资项目 i 的钱，$\mathrm{i} = 1,2, \dots, \mathrm{k}$。目标函数可定义为

$$\max\{g_1(x_1) + g_2(x_2) + \cdots + g_k(x_k)\}$$

$$\text{s.t.} \quad x_1 + x_2 + \cdots + x_k = n$$

定义$G_q(x)$为 x 万元钱投资给前 q 个项目的最大收益。给出如下的递推方程和边界条件

$$G_q(x) = \max\{g_q(x_q) + G_{k-1}(x - x_q)\}, q > 1$$

$$G_1(x) = g_1(\mathrm{x})$$

按照上述递推方程，得到的备忘录表有 n 行 k 列，一共有 nk 项目，按照递推公式，对备忘录中所有的加法次数和比较次数进行统计，时间复杂度为$\mathrm{O}(kn^2)$。

按照递推公式列出下表

x	$G_1(x)$	$G_2(x)$	$G_3(x)$	$G_4(x)$
0	0(0)	0(0,0)	0(0,0,0)	0(0,0,0,0)
1	2(1)	12(0,1)	12(0,1,0)	21(0,0,0,1)
2	7(2)	14(1,1)	16(0,1,1)	33(0,1,0,1)
3	12(3)	19(2,1)	23(0,1,2)	37(0,1,1,1)
4	17(4)	24(3,1)	33(0,1,3)	44(0,1,2,1)

最优解为四个项目分别投资 0、1、2、1。

评分标准：给出动态规划的递推方程及最优解得 7 分（仅给出递推方程得 4 分）；给出时间复杂度得 3 分；推理和计算过程不完整酌情扣分。

2）设计一棵排列树，对该案例而言，投递第 2 个项目有 5 个叶节点，分别表示投资 0、1、2、3 和 4；接下来投资第 2 个项目（假设投资第一个项目 i），则第二个项目有 0，1，…，4-i 个选择，以此类推。

通过回溯算法，若金额等于或者大于 4 万元，则回溯到上一层叶节点。在这一过程中，先得到的第一个投资方案初始化为问题最优解，在后续搜索过程中，采取分支限界的思想来优化最优解。

算法代码略。

评分标准：绘制上述案例的状态空间图（5 分）；给出回溯、分支和限界的条件（5 分）；给出程序源代码或者伪代码（5 分）；不完整酌情扣分。

4. 红果园就餐问题（20 分）

n 个同学去红果园就餐，第 i 名同学所点菜品需要用时为t_i（正整数），其要求完成时间为e_i（正整数）。假定 n 名同学的菜品需要完成一个才能开始下一个（中间不允许出现时间间隔）。如何对 n 名同学进行安排，才能使最大延迟达到最小。

例：小李、小王和小张同学去红果园餐厅吃饭，三位同学所点餐品用时分别为 1 小时、2 小时和 3 小时，要求完成时间分别为 12、13、13（假设餐厅工作时间为 11:00——22:00）。

按照小李（表示为 1）、小王（表示为 2）和小张（表示为 3）的安排顺序，三位同学的延迟（函数 f）分别为

$$f(1) = 11 + 1 - 12 = 0$$

$$f(2) = 11 + 1 + 2 - 13 = 1$$

$$f(3) = 11 + 1 + 2 + 3 - 13 = 4$$

则按照上述安排顺序，最大延迟为 4 小时。

1）设计一个贪心算法，给出贪心算法思路。（5 分）

2）写出程序或者伪代码，并给出该算法的时间复杂度。（7 分）

2）证明该算法的正确性。（8 分）

参考答案：

1）e_i按照升序排列，且各项安排之间没有时间空闲 （贪心策略完全正确 5 分；贪心策略错误不得分）

2）算法伪代码（按照完成时间升序排列安排任务，确保活动间没有时间间隔）

输入：A，T，E　　%A 表示同学集合，T 表示所点菜品需要时间集合，E 表示完成菜品的时间集合

输出：f　% 同学安排的结果

1：排序 A，使得$e_1 \le e_2 \le \cdots \le e_n$

2：f(1) ← 0　　% 设第一个活动从 0 时刻开始

3：i ← 2

4：while i ≤ n　do

5：　　f(i) ← f(i − 1) + t_{i-1}　　%保证所有安排，中间没有时间间隔

6：　　i ← i + 1

给出完整正确的代码或者伪代码，得 7 分；不完善酌情扣分。如果贪心策略错误，不得分。

3）通过交换论证加以证明（贪心算法错误，此处不得分。贪心算法正确，证明没有逆序、没有空闲时间的安排具有相同的最大延迟，得 3 分；证明交换相邻逆序，得到的解仍不降低最大延迟，得 3 分；证明存在有限次交换，得 2 分）。