装
订
线
内

不
要
答
题

北京大学信息科学技术学院考试试卷

考试科目： 算法设计与分析 **姓名：**____________ **学号：**____________

考试时间：2018 年 4 月 18 日 **大班教师:**____________**小班教师:**____________

题号	一	二	三	四	五	六	七	**总分**
分数								
阅卷人								

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

答题要求：解答算法设计题目时，请先用一段话描述算法思想。若用动态规划算法，请写出递推方程、边界条件、标记函数等设计要素；贪心法需给出证明；回溯法需给出解向量、搜索树等、约束条件；各种算法需分析时间复杂度。阅卷时会根据算法的正确性和效率评分。

得分

一、（10 分）填空题（每空 1 分）

1．请将下列 5 个关于 n 的函数 $(\sqrt{2})^{\lg n}$，$n^2, \lg^2 n$，$(\lg n)!$，$2^{\sqrt{2\lg n}}$ 按渐进增长的关系排序，使得 $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \dots, g_4 = \Omega(g_5)$。

$(\lg n)!, n^2, (\sqrt{2})^{\lg n}, 2^{\sqrt{2\lg n}}, \lg^2 n$ 。

2. 对于 n 个元素的数组，插入排序、归并排序和快速排序的最坏情形运行时间分别是_____$O(n^2)$______、_____$O(n\log n)$______和_____$O(n^2)$______。

3．对于某种快速排序算法，如果每两次连续选取的划分元素，一次会把数组划分为基本等长的两个部分，另一次则会划分为长度为 1 和 n-2 的两个部分（n 表示当前数组长度）。则该快速排序算法的时间复杂度是_____$O(n\log n)$______。

4．如果图 $G=(V, E)$中的每条边的长度均为 1，则求给定起点的单源最短路径问题的时间复杂性为____$O(|V|+|E|)$__________。

得分

二、（10 分）求解递推方程。要求给出求解过程或依据。每题 5 分。

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

$T(1) = 1$

答案：

a = 2, b=4, f(n)=n，所以 af(n/b)=2f(n/4)=n/2<=cn=f(n)/2，当 c<1. (2 分)

$n^{\log_b a} = n^{1/2}$，f(n) = n= $\Omega(n^{1/2+\varepsilon})$，$\varepsilon > 0$.　（2 分）

由主定理第三种情况推出

T(n) = Θ(n)　　（1 分）

(2) $T(n) = 2T(n/2) + n/\log n$,

$T(1) = 1$

)　Once again we will begin by expanding the recurrence:

$$\begin{aligned} T(n) &= 2T(n/2) + n/\log n \\ &= 4T(n/4) + 2\frac{n/2}{\log(n/2)} + n/\log n \\ &= 4T(n/4) + \frac{n}{\log\ n - 1} + n/\log n \end{aligned}$$

It is easy to see that the recursion will have $\log n$ levels where each one will cost $\frac{n}{\log(n-i)}$. Thus the sum will be:

$$\begin{aligned} T(n) &= \sum_{i=0}^{\log n} \frac{n}{\log(n-i)} \\ &= n \sum_{i=0}^{\log n} \frac{1}{\log(n-i)} \\ &= n \sum_{i=1}^{\log n} \frac{1}{i} \\ &= \Theta(n \log\log n) \end{aligned}$$

评分标准：　给出递推过程得 2 分，
列出正确的求和公式得 2 分，
最后结果正确得 1 分。

得分

三、（10 分）下面是算法 Find 的伪码，输入 S 是 n 个不等的数构成的数组。请问该算法的输出是什么？该算法在最坏情况下执行了多少次比较运算？

算法 Find(S, n)

```
1.  if S[1] > S[2]
2.  then x←S[1]; y←S[2]
3.  else y←S[1]; x←S[2]
4.  for i←3 to n do
5.      if S[i] > y
6.      then if S[i] > x
7.          then y←x; x←S[i]
8.          else y←S[i]
9.  return y
```

参考答案：该算法的输出是 S 中第二大的数。（5 分）

在最坏情况下第 5 行和第 6 行的比较各执行了 n-2 次，再加上第 1 行的比较，该算法共执行了 2n-3 次比较运算。（5 分）

得分

四、（15 分）矩阵上的快速查找问题

假设有一个 n 维的方形矩阵 $a(1..n, 1..n)$，满足 $a(i, j) < a(i, j+1)$，$a(i, j) < a(i+1, j)$。现已经将矩阵读入内存，需要在矩阵中查找某一个特定的数 x 的位置。设计尽可能快的查找算法完成这个任务。（注：x 可能不出现在方阵中）

（1）写出查找算法，伪码或文字描述均可（5 分）。

方法一：二分搜索分治（满分）

1） 对角线上的元素值为单调递增的；二分法可确定 x 在 $a[k,k]$和 $a[k+1,k+1]$之间

2） 递归的在 $a[1..k,k..n]$和 $a[k..n,1..k]$两个子矩阵上查找 x，输出找到的位置

方法二：直接三分分治（减 1 分）

每次取方阵中点，判断与 x 的大小关系，可划分为三个子问题递归求解。

正确性显然。

方法三：对每行或每列用二分查找（减 1 分）

方法四（满分）：沿着以大小为 x 将矩阵分为左上和右下两块的“分界线”进行查找.

从左下往右上，$a(n, 1)$开始。搜索过程中 $a(i, j)$位置执行以下的搜索策略：

1. 如果 $a(i, j) < x$：此 i 列中无 x，向右移动至 $a(i+1, j)$
2. 如果 $a(i, j) = x$：此 i 列找到 x，记录此时位置，向右移动至 $a(i+1, j)$
3. 如果 $a(i, j) > x$：从当前位置开始向上查找 x，向上移动到 $a(i, j-1)$

从右上到左下，$a(1,n)$开始。搜索过程中 $a(i, j)$位置执行以下的搜索策略：

1. 如果 $a(i, j) < x$：从当前位置开始向下查找 x，向下移动到 $a(i, j+1)$
2. 如果 $a(i, j) = x$：此 i 列找到 x，记录此时位置，即向左移动至 $a(i-1, j)$
3. 如果 $a(i, j) > x$：此 i 列中无 x，向左移动至 $a(i-1, j)$

后面两小题（证明和算法复杂度）只考虑对其所选算法的分析是否正确。

（2）分析算法的正确性（5 分）

方法一：

$a(i,j) < a(i,j+1)$，$a(i,j) < a(i+1,j)$可推出 $a(i,j) < a(i+1,j+1)$，故从左上到右下的对角线上的元素是严格单调递增的。因此二分法可以确定 x 在某对$[a(k,k), a[k+1,k+1]]$之间。再根据 $a(i,j) < a(i,j+1)$，$a(i,j) < a(i+1,j)$，可排除 x 在方阵 $a(1..k, 1..k)$和 $a[k+1..n,k+1..n]$上的情形。而 $a[1..k,k..n]$和 $a[k..n,1..k]$两个子矩阵上查找 x 恰好是原问题规模更小的子问题，可以用递归的方式求解，当子矩阵足够小时，直接给出结果。

注：只要能说明把原问题分治为两个子问题即可证明算法的正确性。

方法二：分析类似方法 1，只要能说明把原问题分治为三个子问题即可证明算法的正确性。

方法三：因为每行和每列都是排好序的，所以可以用二分查找。

方法四：（右上到左下为例）重点是说明在第 i 列到第 i-1 列的移动过程，采用的搜索策略仍然能保证找出正确解，不会漏解。

假设在 $a(i,j)$位置处出现 $a(i,j) >= x$，需要向左移动至 $a(i\text{-}1,j)$。此时只需要说明 $i\text{-}1$ 列中 j 行以上，$a(i, 1\sim j\text{-}1)$中一定不存在 x 即可，即只需要说明 $a(i\text{-}1,j\text{-}1) < x$。

因为 $a(i\text{-}1,j\text{-}1) < a(i,j\text{-}1)$，而能够到达 $a(i,j)$就说明 $a(i,j\text{-}1) < x$，所以 $a(i\text{-}1,j\text{-}1) < x$。所以这种搜索策略可以不漏解的找出所有 x。

（3）说明算法的时间复杂度（5 分）。

方法一：$T(n) = 2T(n/2) + \log n = O(n)$ 或者 $O(n\log n)$ 都对

方法二： $T(n)=3T(n/2) + O(1) = O(n^{\log_2 3})$

方法三： $T(n) = n\log n$

方法四：$T(n) = O(n)$

说明：对于其他解法，如果能证明其正确性，复杂度能达到 O(n)，可以得满分；复杂度达到 O(nlogn)，可以得 14 分。

得分

五、（20 分）五一假期，小江同学计划开车从 A 地出发，到相距为 L 公里的 B 地探险。在 A 地到 B 地的途中有 n 个加油站，它们分别分布在距离 A 地 d_i ($i = 1, 2, \dots, n$)公里的位置，每个加油站可以提供 V_i 升的汽油。一开始时小江同学的汽车上有 S 升汽油，汽车每行驶一公里需要耗油一升，假设汽车油箱容量无穷。

请问小江同学在途中至少要加几次油？她要在哪些加油站加油？

请设计一个算法解决上述问题，写出算法描述，分析算法的时间复杂度，并证明算法的正确性。

参考答案：使用贪心法。

（1） 贪心策略：从起点向终点扫描，一旦遇到没有油的情况，则选择已经扫描过的加油站中油量最多的加油站进行加油。 （5 分）

（2） 时间复杂度：每次选择时都需要维护已扫描加油站油量的最大值，时间复杂度为 O(log n)，最坏情况下会选择 n 次加油站，总时间复杂度为 O(n log n)。（5 分）

（3）正确性证明：使用交换论证的方法。任给一组最优解，其中必然存在一个加油站 t_1 位于[0, S]中，如果贪心法选择的第一个加油站 T_1 不在解中，则有 $V_{T1}>V_{t1}$，用 T_1 替换 t_1，不影响解的正确性。同理，在[0, S+V_{T1}]中除 T_1 外至少存在一个加油站 t_2，如果它不是贪心法选择的第二个加油站 T_2，则有 $V_{T2}>V_{t2}$，用 T_2 替换 t_2，不影响解的正确性。以此类推，可以将这组最优解替换成贪心法的解。（10 分）

得分

六、（20 分）五一假期，小江同学计划开车从 A 地出发，到相距为 L 公里的 B 地探险。在 A 地到 B 地的途中有 n 个加油站，它们分别分布在距离 A 地 d_i $(i = 1, 2, \ldots, n)$公里的位置，每个加油站可以提供 V_i 升的汽油。一开始时小江同学的汽车上有 S 升汽油，汽车每行驶一公里需要耗油一升，假设汽车油箱容量无穷。

小江同学希望在从 A 地和 B 地的途中用汽油换购一些土特产。在 A 地到 B 地的途中有 m 个售卖土特产的商店，它们分别分布在距离 A 地 e_j $(j = 1, 2, \ldots, m)$公里的位置。在每一个商店，小江同学可以使用 p_j 升的汽油换购一件价值为 c_j 的商品，或者选择什么都不换购。在小江同学能够顺利到达 B 地的前提下，请问她最多能换购多少价值的土特产？

参考答案：使用动态规划算法。

先从终点向起点扫描，假设在每个加油站都加满油，预处理得到在每个商店处可以用于换购商品的最大油量 H_j $(j = 1, 2, \ldots, m)$。

目标函数：$\max \sum_{j=1}^{m} c_j x_j$,

$$\text{s.t. } \sum_{j=1}^{k} p_j x_j \leq H_k (k = 1, 2, \ldots, m),\ x_j = 0, 1。$$

递推方程：设 R[k, y]是汽车到达第 k 个商店，花费 y 升汽油所能得到的最大价值，那么有

$$R[k, y] = \begin{cases} R[k-1, y] & y < p_k \wedge y \leq H_k \\ \max\{R[k-1, y], R[k-1, y-p_k] + c_k\} & p_k \leq y \leq H_k \\ -\infty & y > H_k \end{cases}, k > 1$$

边界条件：$R[1, y] = \begin{cases} 0 & y < p_1 \wedge y \leq H_1 \\ c_j & p_1 \leq y \leq H_1 \\ -\infty & y > H_1 \end{cases}$

时间复杂度：预处理复杂度为 O(n+m)，动态规划复杂度为$O(m \cdot (S - L + \sum_{i=1}^{n} V_i))$，总复杂度为$O(m \cdot (S - L + \sum_{i=1}^{n} V_i))$。

得分

七、（15 分）有了五一假期的探险经验，小江同学计划在暑假期间去更多的地方探险。小江同学的目标地点有 n 个，分别为 A_1, A_2, …, A_n，其中每个地点都有一个可以提供 V_i 升油的加油站，任意两个地点 A_i 和 A_j 之间有一条长度为 d_{ij} 的道路。一开始时小江同学的汽车上有 S 升汽油，汽车每行驶一公里需要耗油一升，假设汽车油箱容量无穷。

小江同学希望能从某一地点出发，在不重复地经过所有其它地点之后回到这一地点，请问她能够完成这一心愿吗？

参考答案：使用回溯算法。

解向量：<$x_1, x_2, \ldots, x_n$>，表示汽车依次经过的地点，部分向量<$x_1, x_2, \ldots, x_k$>，表示前 k 个经过的地点。

搜索树：树高为 n+1 的排列树，树中关联根节点的第 1 层的边有 n 条，表示起始地点的编号，第 2 层有 n-1 条边，分别表示下一步要到达的地点编号，以此类推。

搜索策略：深度优先，在第 k 层的节点处（根节点为第 0 层）计算汽车的剩余油量 $R_k = S + \sum_{i=1}^{k-1}(V_i - d_{i,i+1})$，如果油量为负则回溯。

时间复杂度：最坏情况下为 O(n!)