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： 2015 年 4 月 27 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

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、（15 分）按照阶递减的顺序排列下面的函数。如果函数 $f(n)$与 $g(n)$的阶相同，就表示成 $f(n)=\Theta(g(n))$，本题只需要给出结果。

$$2^{\sqrt{2\log n}},\quad n\log n,\quad \sum_{k=1}^{n}\frac{1}{k},\quad n2^n,\quad (\log n)^{\log n},\quad 2^{2n},\quad 2^{\log\sqrt{n}}$$

$$n^3,\quad \log(n!),\quad \log n,\quad \log\log n,\quad n^{\log\log n},\quad n!,\quad n,\quad \log 10^n$$

答案：

$$n!,\quad 2^{2n},\quad n2^n,\quad (\log n)^{\log n}=\Theta(n^{\log\log n}),$$

$$n^3,\quad \log(n!)=\Theta(n\log n),\quad \log 10^n=\Theta(n),\quad 2^{\log\sqrt{n}},\quad 2^{\sqrt{2\log n}}$$

$$\sum_{k=1}^{n}\frac{1}{k}=\Theta(\log n),\quad \log\log n$$

得分

二、（10 分）求解下述递推方程，其中 k 是给定的正整数．要求给出求解过程.

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

$$T(1)=1$$

解答：

$$T(n)=O(n^2\log^{k+1}n\ )$$

或

$$T(n)=\Theta(n^2\log^{k+1}n\ )$$

得分

三、（15 分）

设 A 是 n 个实数的数组，考虑下面的递归算法：

```
XYZ (A[1..n])
if   n=1 then return A[1]
else   temp ← XYZ (A[1..n−1])
          if   temp < A[n]
          then   return   temp
          else   return   A[n]
```

1. 用简短的文字说明算法 XYZ 的输出是什么？
2. 以 A 中元素的比较作为基本运算，列出算法 XYZ 最坏情况下时间复杂度 $W(n)$ 的递推方程，并解出 $W(n)$。
3. 在求解这个问题的算法类中，算法 XYZ 最坏情况下是不是效率最高的算法？为什么？

解答：

1. A 中的最小实数。

2. $W(n)=W(n-1)+1$

   $W(1)=0$

   $W(n)=n-1$

3. 是效率最高的算法，因为找最小问题至少需要比较 $n-1$ 次。

得分

四、（15 分）设 A 是 n 个数的序列，如果 A 中的元素 x 满足以下条件：小于 x 的数的个数≥ $n/4$，且大于 x 的数的个数≥$n/4$，则称 x 为 A 的近似中值．设计算法求 A 的一个近似中值．说明算法的设计思想和最坏情况下的时间复杂度.

答案

算法 1：

1. 用 Select 算法找第$\lceil n/4 \rceil$小的数 a 和第$\lfloor 3n/4 \rfloor$小的数 b
2. if a=b return “无解”
3. else 用 a 和 b 划分数组 A 为 A_1, A_2, A_3, A_4, A_5，其中 A_1 的数<a，A_2 的数=a,
   A_3 的数>a 且小于 b，A_4 的数=b. A_5 的数>b. （当 a=b 时，无解）
4. if A_3 非空，则 A_3 中的数为近似中值，否则无解.

时间 $O(n)$.

得分

五、（15 分）在一组服务器$S = \{s_1, s_2, \ldots, s_n\}$上放置文件副本，如果副本放置在$s_i$上，则产生$c_i$（正的整数值）的存储代价．当在$s_i$上发生对该文件的访问时，如果文件副本在$s_i$上，则无访问代价；如果不在$s_i$上，则需要顺序查找$s_{i+1}, s_{i+2}, \ldots, s_j$，直到在$s_j$上找到文件副本，这将产生$j - i$的访问代价．规定副本至少一定要放置在$s_n$上，以便所有的访问均能成功．问当每个服务器上均发生对该文件的一次访问时，存储代价加访问代价之和最少是多少？设计一个算法计算之，说明算法的正确性，并分析算法的时间复杂度．

参考答案一：

$$F_n[k] = \begin{cases} 0 & k = n \\ \min\left\{F_n[j] + c_j + \dfrac{(j-k)(j-k-1)}{2} \,\middle|\, k < j \le n\right\} & 0 \le k < n \end{cases}$$

$F_n(k)$代表仅考虑在$s_{k+1}, s_{k+2}, \ldots, s_n$这些服务器上放置文件副本的最小总代价．求$F_n(0)$．

递推式的含义为，枚举第一个副本放置的服务器编号，计算该副本承担的代价，加上该副本之后的服务器上最优副本放置方案的代价，取最小值为最优结果．

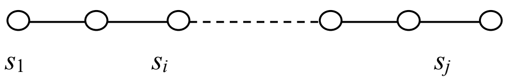
$$F_n[l] = \begin{cases} 0 & l = 0 \\ \min\left\{F_n[t] + c_{n-t} + \dfrac{t(t-1)}{2} \,\middle|\, 0 \le t < l\right\} & 0 < l \le n \end{cases}$$

$F_n[l]$表示只考虑后 l 个服务器上如何放置文件副本的子问题。求$F_n[n]$。

参考答案二：

关于副本放置问题，有一个比较简单的表示，只需要一个参数来界定子问题。

令 OPT[j]表示副本放在 s_j，从 s_1 到 s_j 的最小代价，那么原始问题求的是 OPT[n]。



假如在 s_j前面，离 s_j最近的放置副本的位置是 s_i，那么

$$\mathrm{OPT}[j] = c_j + \min\{\mathrm{OPT}[i] + C(i+1, j) \mid 0 \le i < j\}$$

其中 $C(i+1, j) = (j-i)(j-i-1)/2$

$$OPT[0]=0$$

上式的 $C(i+1,j)$ 代表 s_{i+1} 到 s_{j-1} 这些服务器上的访问代价之和，其中 s_{j-1} 的代价是 1，s_{j-2} 的代价是 2，…， s_{i+1} 的代价是 j-i-1. 求和是

$$1+2+\cdots+(j-i)=(j-i)(j-i-1)/2$$

得分

六、（15 分）一个公司需要购买 n 个密码软件的许可证，按规定每个月至多可得到一个软件许可证. 每个许可证当前售价都是 1000 元，但是第 i 个许可证的售价将按照 r_i >1 的指数因子增长，i=1, 2, ⋯, n. 例如，第 i 个许可证的售价在 1 个月后将是 r_i×1000 元，2 个月后将是 r_i^2×1000 元，k 个月后将是 r_i^k ×1000 元. 假设 r_1, r_2, ⋯, r_n是给定正整数，试给出一个购买许可证的顺序，以使得花费的总钱数最少. 设计一个算法求解这个问题，说明设计思想，证明其正确性并分析算法最坏情况下的时间复杂度.

答案

排序 r_i为递减次序，使得 $r_1 \geq r_2 \geq \cdots \geq r_n$，依次购买.

设最优解为 OPT(I)，假设在 OPT(I)存在逆序，即 $r_j < r_i$，但是 j 在 i 前面购买. 一定有相邻的逆序，即存在 i 和 j，使得 j 在 i 前面与 i 相邻. 设 j 是第 t 个月购买，i 是第 $t+1$ 个月购买. 那么交换 i 与 j 得到解 $S(I)$，那么花费之差

$$V(\mathrm{OPT}(I)) - V(S(I))$$
$$=(r_i^{t+1}\times 1000 + r_j^{\,t}\times 1000) - (r_i^{\,t}\times 1000 + r_j^{t+1}\times 1000)$$
$$=[\,r_i^{\,t}(r_i-1) - r_j^{\,t}(1-r_j)\,]\times 1000$$

由于 $r_i \geq r_j$，于是 $r_i-1 \geq r_j-1$ 且 $r_i^{\,t} \geq r_j^{\,t}$，得到上式 ≥ 0.

时间为 $O(n\log n)$.

得分

七、（15 分）某会展中心有 *m* 行 *n* 列会议室阵列，每间会议室为边长为 10 米的正方形，现在要在会议室中设置无线路由器，假设只能在会议室的中心位置放置路由器，每个路由器的信号覆盖范围是一个半径为 25 米的圆形，请问如何设计路由器的位置，使得所有的会议室的中心点都有信号并使得路由器个数最少。

答案:

解向量为 mn 维<x_11, x_12, …, x_mn>， x_ij=1 表示会议室(i, j)放置路由器，x_ij=0 表示不放。这样搜索树为 mn 层。

初始令所有的会议室都放置路由器，算法从会议室（1,1）开始，左子树表示拿走（1，1）会议室的路由器，右子树表示保留。这样可证明满足多米诺性质。在进入左子树时要检查是否有会议室未被覆盖，若有则停止搜索该分支。

检查涉及的会议室个数为常数个，每个涉及的会议室只需要寻找能否在 25 米之内找到路由器，这个也是常数时间。所以总的时间复杂度最坏情况为 O(2^mn)。