

2024-2025 学年第一学期 算法设计与分析 期末考试卷 (A 卷)

一、选择题 (单项选择, 每题 2 分, 共 20 分)

1. 如果有两个正常数 c 和 n_0 , 对于所有的 $n \geq n_0$, 有 $|f(n)| \leq c|g(n)|$, 则记作【 】。
A. $f(n)=O(g(n))$
B. $f(n)=o(g(n))$
C. $f(n)=\Omega(g(n))$
D. $f(n)=\theta(g(n))$
2. 算法的存储量不包括【 】。
A. 输入数据所占空间
B. 算法本身所占空间
C. 辅助变量所占空间
D. 问题的规模
3. 归并排序法是【 】 算法策略的应用。
A. 枚举算法
B. 贪婪算法
C. 分治算法
D. 动态规划
4. 算法策略主要有: 分治法、贪心法、动态规划法、【 】 和分支限界法等。
A. 微分法
B. 积分法
C. 回溯法
D. 筛选法
5. 算法的复杂性主要依赖于【 】。
A. 算法实现的平台
B. 算法的输入

- C. 要解决问题的规模
- D. 设计者的学术水平
6. 用蛮力法搜索子集树时，算法的复杂度为 【 】。
- A. $O(n^2)$
- B. $O(2^n)$
- C. $O(n!)$
- D. $O(n \log_2 n)$
7. 下列需要具有最优子结构性质的算法策略是 【 】。
- A. 概率算法
- B. 回溯法
- C. 分支限界法
- D. 动态规划法
8. 算法的三要素包括 【 】。
- A. 数据、控制结构、操作
- B. 操作、控制结构、数据结构
- C. 逻辑结构、存储结构、数据结构
- D. 顺序、选择、循环
9. 下列程序段的功能是 【 】。
- ```
void fun(int n) {
 if (n < 10) printf("%d", n);
 else { fun(n/10); printf("%d", n%10); }
}
```
- A. 从高位到低位逐位输出一个十进制整数的各位数字
- B. 从低位到高位逐位输出一个十进制整数的各位数字
- C. 输出一个十进制整数的各个因子
- D. 输出一个十进制整数的各位互不相同的数字

10. 利用递归算法解题的一般步骤为 【     】。

①分析问题，寻找递归关系；②设置边界，控制递归；③设计函数、确定参数；  
④从具体到抽象设计数据结构

A. ①②③

B. ①④②③

C. ①②③④

D. ④①②③

二、分析简答题 (本大题共 8 小题，每小题 5 分，共 40 分)

11. 递归方程求解：若当  $n=1$  时， $T(n)=1$ ，请计算当  $n>1$  时问题的复杂度  
 $T(n)=2T(n/2)+2$ 。

12. 与算法执行时间相关的因素有哪些？

13. 在某些情况下，需要处理很大的整数，设  $X$  和  $Y$  都是  $n$  位的二进制整数，现在要计算它们的乘积  $X \times Y$ 。请用“分治法”设计一个有效的大整数乘积算法（简述思路）。

14. 适用“动态规划”策略解决的问题的特征是什么？

15. 什么是“割点”？简述连通图中割点的判别方法。

16. 比较“回溯法”和“分支限界法”，它们之间的不同点有哪些？

17. (代码阅读) 请阅读下列算法，若输入  $n$  的值为 5，则写出代码段的执行结果。

```
int main() {
 int i, j, a[10][10], n, k;
 scanf("%d", &n);
```

```

k=1;

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

 for(j=1; j<=n+1-i; j++)

 { a[i-1+j][j]=k; k++; }

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

 printf("\n");

 for(j=1; j<=i; j++) printf("%-3d", a[i][j]);

}

return 0;

}

```

18. (代码分析) 执行下列程序段时输入 n 个数据，请分析该程序段的功能。

```

#include <stdio.h>

int main() {

 int a[100], i, j, t=1, n;

 scanf("%d", &n);

 for(i=1; i<=n; i++) scanf("%d", &a[i]);

 for(i=1; i<=n-1; i++)

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

 if(a[i]==a[j]) { t=0; break; }

 if(t==1) printf("Non!\n");

 else printf("Yes!\n");

 return 0;

}

```

四、算法填空题 (本大题共 2 小题, 每空 2 分, 共 20 分)

19. [取数游戏] 有 2 个人轮流取  $2n$  个数中的  $n$  个数, 所取数之和考虑到大者为胜。下列算法的作用是, 让先取数者胜, 模拟取数过程。请在空格处将算法补充完整。

```
#include <stdio.h>

int main() {

 int i, s1, s2, data, n;

 scanf("%d", &n);

 (1) _____;

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

 scanf("%d", &data);

 if ((2) _____)

 (3) _____;

 else

 (4) _____;

 }

 if ((5) _____) ...

}
```

20. [高精度数搜索] 寻找一个数, 使得该数的前  $i$  位 (即高  $i$  位) 能被  $i$  整除。

```
// 省略部分头文件

for(i=2; i<=100; i++) A[i]=0; // 置初值

A[1]=1;

n=1; i=1;

while((1) _____) // 循环条件

{

 if (i >= n) // 发现有更大的满足条件的高精度数据
```

```

{
 (2) _____; // 更新 n 的值
 for (k=1; k<=n; k++) B[k]=A[k]; // 转存到数组 B 中
}

i=i+1; r=0;

for(j=1; j<=i; j++) // 检查第 i 位是否满足条件
{
 (3) _____; // 计算余数
 r = r % i; // r = r mod i
}

if (r != 0) // 若不满足条件
{
 (4) _____; // 当前位数字加 1
 while (A[i]>9 && i>1) // 搜索完第 i 位的解, 回溯到前一位
 {
 A[i]=0;

 (5) _____; // 回溯

 A[i]=A[i]+1; // 上一位加 1
 }
}
}

```

五、算法设计题 (本大题共 2 小题, 每小题 10 分, 共 20 分)

21. 利用“递归算法”设计: 找出  $n$  个自然数  $(1, 2, 3, \dots, n)$  中取  $r$  个数的组合, 并统计组合的总数。

22. 设  $f(x)$  是  $[0, 1]$  上的连续函数, 且  $0 \leq f(x) \leq 1$ 。设计一个通用的“蒙特卡罗随机算法”: 计算积分  $\int_0^1 f(x)dx$ , 积分值等于图中的面积  $G$ 。

### 【附录：参考答案与解析】

#### 一、选择题

1. A (大 O 定义)
2. D (问题规模不是算法本身存储)
3. C (归并排序是分治)
4. C (四大策略之一)
5. C (复杂度依赖规模)
6. B (子集树  $2^n$ )
7. D (动态规划特征)
8. B (算法三要素)
9. A (递归正序输出)
10. A (递归设计步骤)

#### 二、分析简答题

11.  $T(n) = O(n)$ 。(根据主定理,  $a=2, b=2, d=0$ , 属于 case 1, 但常数项累加, 实际每层  $O(1)$ , 共  $n$  层)。
12. 硬件速度、编译程序代码质量、算法策略、问题规模。
13. 思路: 将  $n$  位大整数  $X, Y$  拆分为高低两部分, 利用公式  $XY = AC \cdot 2^n + ((A-B)(D-C) + AC + BD) \cdot 2^{n/2} + BD$ , 将乘法次数从 4 次减少为 3 次。
14. 两个特征: 最优子结构性质、子问题重叠性质。
15. 割点: 删除某点及相关边后图不连通。判别: DFS 中根节点有  $\geq 2$  子树; 非根节点  $u$  的子节点  $v$  无法回溯到  $u$  的祖先。
16. 回溯法: DFS 搜索, 找所有解, 空间小; 分支限界法: BFS 搜索, 找最优解, 需维护活结点表, 空间大。
17. 输出下三角矩阵形式:

1

6 2

9 7 3

11 10 8 4

12 ... (斜向填充逻辑)

18. 功能：判断输入的  $n$  个数中是否有重复数字。有重复输出 Yes!，无重复输出 Non!。

#### 四、算法填空题

19. (1)  $s1=0, s2=0$ ; (2)  $s1 \leq s2$  (3)  $s1 += data$ ; (4)  $s2 += data$ ; (5)  $s1 > s2$

20. (1)  $i > 0$  (2)  $n = i$  (3)  $r = r * 10 + A[j]$ ; (4)  $A[i]++$ ; (5)  $i--$ ;

#### 五、算法设计题

21. 核心递归逻辑 Combine( $n, r$ ):

if ( $r=0$ ) { Print(); return; }

if ( $n < r$ ) return;

Combine( $n-1, r-1$ ); // 选  $n$

Combine( $n-1, r$ ); // 不选  $n$

22. 蒙特卡罗投点法:

在  $1 \times 1$  矩形内随机产生  $N$  个点  $(x_i, y_i)$ ;

统计满足  $y_i \leq f(x_i)$  的点数  $m$ ;

积分值  $\approx m / N$ 。