

22年算法考试

1、单选题

1、算法的执行时间和原操作执行次数之和（C）

第二章 算法的时间复杂性 时间复杂度估算 书本P37

A、无相关性

B、成反比例关系

C、成正比关系

D、的平方成正比比例关系

2、衡量算法优劣的标准通常为（B）

第二章 算法分析的评价体系 P36

A、算法的可读性

B、算法的时间复杂性和空间复杂性

C、实现算法代码量的多少

D、算法的正确性

3、下列关于算法描述不正确的是（D）

A、算法是计算机解决问题的过程

p5 算法机器要素和特征 算法的定义

B、算法由操作、控制结构、数据结构三要素组成

p5 算法的三要素 算法的3要素

C、算法操作的对象就是数据

p5 数据结构

D、关系运算是操作的一种类型

关系比较

4、算法描述 p12

- A、自然语言是无法描述算法
- B、源程序是表示算法最好的方式
- D、算法描述中最难用的方法是流程图

5、适合解决残缺棋盘问题的算法是 (B)

p140

- A、蛮力法
- B、分治法
- C、动态规划法
- D、迭代法

6、下列关于循环和递归的比较说法正确的是 (C)

p51 循环与递归

- A、凡是能用循环编写的算法，都可以用递归算法实现
- B、递归算法总是比对应的循环算法执行速度快
- C、从程序的可读性上来讲，非递归要比递归更容易理解
- D、从占用空间角度来讲，递归程序运行时占用空间更小

7、有关算法渐进时间复杂度数量级的说法不正确的是 ()

- $O(1)$ 称为常数级。
- $O(\log n)$ 称为对数级。
- $O(n)$ 称为线性级。
- $O(n^c)$ 称为多项式级。
- $O(c^n)$ 称为指数级。
- $O(n!)$ 称为阶乘级。

8、

定义 1: 如果存在两个正常数 c 和 n_0 , 对于所有的 $n \geq n_0$, 有 $|f(n)| \leq c |g(n)|$, 则记作 $f(n) = O(g(n))$ 。

定义 2: 如果存在两个正常数 c 和 n_0 , 对于所有的 $n \geq n_0$, 有 $|f(n)| \geq c |g(n)|$, 则记作 $f(n) = \Omega(g(n))$ 。

P38 问题时间复杂度的上界下界

9、“数塔问题”通常采用以下哪种算法求解 (D)

- A、蛮力算法
- B、分治策略
- C、回溯法
- D、动态规划法

10、下列不属于优化算法的基本技巧的是 (A)

看目录

- A、利用并行算法改进算法
- B、利用算数运算简化算法
- C、利用标志量改进算法
- D、信息数字化

2、判断题

11、正确性是算法的基本特征之一，即一个算法必须是正确的 (对)

1.1.3 算法设计及基本方法

算法设计(designing algorithm)作为用计算机解决问题的一个步骤,其任务是对各类具体问题设计出良好的算法。算法设计作为一门课程,是研究设计算法的规律和方法。

在设计算法时,应当严格考虑算法的以下质量指标。

1) 正确性(correctness)

首先,算法对于一切合法的输入数据都能得出满足要求的结果;其次对于精心选择的、典型的、苛刻的几组输入数据,算法也能够得出满足要求的结果。

2) 可读性(readability)

算法主要是为了人的阅读与交流,其次才是让计算机执行。因此算法应该易于人的理解;另一方面,晦涩难读的算法易于隐藏较多错误而难以调试;有些算法设计者总是把自己设计的算法写的只有自己才能看懂,这样的算法反而没有太大的实用价值。

3) 稳健性(robustness)

当输入的数据非法时,算法应当恰当地做出反映或进行相应处理,而不是产生莫名其妙的输出结果。这就需要一定要充分地考虑到可能的异常情况(unexpected exceptions),并且处理出错的方法不应该是简单地中断算法的执行,而应是返回一个表示错误或错误性质的值,以便在更高的抽象层次上进行处理。

4) 高效率与低存储量需求

通常,效率指的是算法执行时间;存储量指的是算法执行过程中所需的最大存储空间。两者都与问题的规模有关。这一点在第2章中详细介绍。

针对不同的问题算法设计方法、策略很多,学习和掌握它们是本书的主要任务,这里先介绍几个算法设计的基本模型。

12、可以在多项式的时间内解决的判定性问题属于P类问题 (对)

p40 NP完全问题

13、算法的时间复杂度是评价算法好坏的准一标准 (错)

14、 $O(N)$ 为多项式级 (错)

15、“贪婪”算法可以理解为以逐步的局部最优。达到最终的全局最优

p151 贪婪算法

16、 $f(n)$ $g(n)$ 都是关于正整数的函数,若存在一个常数 C ,使得 $f(n)/g(n) = C$ 时,称 $f(n)$ $g(n)$ 同数量级的函数 (错)

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = C$$

则称 $f(n)$ 与 $g(n)$ 是同数量级的函数。

17、折半查找算法的空间复杂度为 $O(\log n)$ （对）

18、枚举法是蛮力策略的一种表现形式（对）

19、循环是一种比递归更强、更好用的实现“重复操作”的机制（错）

| p67 反了

20、迭代算法有时不能求出问题的精确解（对）

填空题

21、ANSI中规定的流程图符号中，矩形框表示【加工、处理】

| P13

22、对于图 $G(V, E)$ 而言， n 个顶点， e 条边，如果用邻接表表示图，其广度优先搜索算法的时间复杂度是【 $O(n+e)$ 】

23、下列算法片段的时间复杂度为【 $O(\log_2 n)$ 】

```
1 i = 1;
2 while(i <= n)
3     i = i * 2;
```

24、

定义 1：如果存在两个正常数 c 和 n_0 ，对于所有的 $n \geq n_0$ ，有 $|f(n)| \leq c |g(n)|$ ，则记作 $f(n) = O(g(n))$ 。

定义 2：如果存在两个正常数 c 和 n_0 ，对于所有的 $n \geq n_0$ ，有 $|f(n)| \geq c |g(n)|$ ，则记作 $f(n) = \Omega(g(n))$ 。

25、若算法的两个部分的时间复杂度为 $O(f(n))$ 和 $O(g(n))$ ，则总的时间复杂度为

【 $O(\max(f(n), g(n)))$ 】

26、求解百鸡百钱问题，可以使用蛮力法，可以通过优化使得枚举仅需 $20 * 33 = 660$ 次即可，此算法的时间复杂度为：【 $O(n^2)$ 】

27、用递归算法计算 $n!$ 的时间复杂度为【 $O(n)$ 】

28、没有环且每两个顶点间最多只有一条边相连的图称为【简单图】

| p189

29、下面片段的时间复杂度为【 $O(\log(2)n)$ 】

```
1 for (i = 0; i < N ; i *= 2)
2     for( j = 0; j < 100; j++ )
3         for(k = 0; k < 100; k++)
4             { ..... }
```

30、常见的几种分支搜索算法中有：先进先出搜索算法、【优先队列式搜索算法】、（LIFO）搜索算法和优先级搜索算法

主观题

31、狼找兔子问题：一座山周国有20个洞，顺时针编号为0，1，2，3，4。。。18，19。有一只狼从0号洞开始找，顺时针方向计数，它每找一个洞就间隔几个再找下一洞，间隔的数依次为1，2，3，。。。当狼找了10个洞后就累的走不动，不找了。问兔子在哪些洞里时可以幸免？请编写算法求解。

```
1 int tmp = 0, i = 0;
2 visited[20];
3 for(i; i < 10; i++){
4     tmp=(tmp + i) /20;
5     visited[tmp] = 1;
6 }
7
8 for(j = 0; j < 20; j++){
```

```

9      if(visited[j] == 0)
10         println(j)
11 }

```

32、输入n个元素的一个一维数组，如果将其从小到大排序，求原第1个元素应该在排序后数组的第几个位置？编写不用排序也能求出问题解的算法。

```

1 int find (int a[], n) {
2     cnt = 0;
3     for(i = 1; i < n; i++){
4         if(a[i] < a[0]){
5             cnt++;
6         }
7     }
8     return cnt;
9 }

```

33、一只小松鼠收集了一堆松果，它要运回树洞里，计划每次运走一半。但是它每次回来时都会发现少了2个松果（因为被另外的松鼠吃了），当它第5次回来运的时候，发现只剩下1个了，问原来有多少个松果？写一个算法进行计算。

```

1 int x = 1;
2 for(i = 0; i < 4; i++){
3     x = (x + 2) * 2;
4 }
5 println(x);

```

34、乌鲁木齐市红山公园从山脚到山顶有n级台阶，人上山时每步可以选择上一级台阶或两级台阶，编写程序计算共有多少种上山的步法。

```

1 long count(n){
2     long a[n+1];
3
4     a[0] = a[1] = 1;
5     for(i = 2; i <= n; i++){
6         a[i] = a[i-1] + a[i-2];

```

```

7     }
8
9     return a[n]
10  }

```

35、利用蒙特卡罗算法，计算函数 $f(x)$ 在 $[0,2]$ 区间上的定积分这里假定 $f(x)$ 是已经定义好的函数，且在区间

内的值小于1。（将 n 根飞镖随机投向一个长方形的靶子计算落入 $y=f(x)$ 与 x 轴之间在 $[0, 2]$ 区间中的飞镖数目 k ，和总飞标数之比，即为二者的面积比。从而可以计算定积分 I 的近似值）

```

1 area(int total){
2     int inside = 0;
3     for(i = 0; i < total; i++){
4         x = fRandom(0,2);
5         y = fRandom(0,1);
6         if(y <= f(x))    inside ++;
7     }
8     return (float)inside / total * 2;
9 }

```

36、 $\cos(x)=1 - x^2/2! + x^4/4! - x^6/6! + \dots + (-1)^n x^{2n}/(2n)! + \dots$ 请编写算法计算 $\cos(x)$ 的近似值，误差小于 10^{-6}

```

1 cos(int x){
2     float num;
3     int i = 1;
4     float result = 1;
5     float tmp = 1;
6
7     do{
8         tmp = -1 * tmp * x * x / ((2*i)*(2*i-1));
9         result += tmp;
10    }while(tmp >= 0.000001)
11
12    return result;
13 }

```

37、写出下图的邻接矩阵，并编写从顶点1开始的广度优先搜索算法

	1	2	3	4	5	6	7
1	0	1	0	1	0	0	0
2	0	0	1	0	1	0	0
3	0	0	0	0	0	1	0
4	0	0	1	0	0	0	0
5	0	0	0	0	0	1	1
6	0	0	0	1	0	0	1
7	0	0	0	0	0	0	0

```

1  int visited[8]
2  int k = 1;
3  bfs(k,grap[8][8],int n = 7){
4      queue Q;
5      Q.push(k);
6      visited[k] = 1;
7      println("visited", k);
8      while(Q.isNotEmpty){
9          tmp = Q.pop();
10         for(i = 1; i <= 7; i++){
11             if(visited[i] == 0 && grap[tmp][i] == 1){
12                 visited[i] = 1;
13                 Q.push(i);
14                 println("visited", i);
15             }
16         }
17     }
18 }

```

```

1  //深度优先搜索
2  int visited[8]
3  main(){
4      dfs(1);
5  }
6  dfs(int k){
7      visited[k] = 1;

```

```

8
9     for(i = 1; i <= 7; i++){
10         if(graph[k][i] == 1 && visited[k] = 0;)
11             dfs(i);
12     }
13 }

```

38、警察局抓了a, b,c,d,e五个嫌疑犯，其中只有一人是小偷。审问中，a说：“小偷是c、d、e中的一个。”，b说：“我不是小偷。”，c说“小偷肯定是d。” · d说：“c在冤枉人。”，e说：“小偷是c”。现在已经知道5个人中有4个人说的是真话一个说的是假话，问到底谁是小偷。请编写一个程序帮警察找出小偷。

```

1  for(x = 'a'; x <= 'e'; x++){
2      if(((x == 'c' || x == 'd' || x == 'e') + (b != x) + (x == d) + (x != d) + (x
3          println("theif is ", x);
4      }
5  }

```

39、下面是“8皇后问题”回溯算法的算法框架，请将划线部分补全。

(1) 用一个一维数组X[8]存储皇后位置,若第行第列放有皇后,则X[i] = j。

(2) 分别用一维数组C[8]、L[16]、R[16]表示竖列、左斜线、右斜线是否放有棋子，有则值为1,否则值为0。

```

1  int X [8],C[8],L[16],R[16];
2  int n, i, j, k;
3
4  void main ()
5  {
6      int i,    n=8:
7      for(i = 0 ; i < 8; i++){
8          x[i] = 0;
9          c[i] = 0;
10         l[i] = 0;
11         l[i+n] = 0;
12         r[i+n] = 0;
13     }
14
15     try(1);

```

```

16 }
17
18 try(int i){
19     if(i >= 8){
20         cnt ++
21         return;
22     }
23     for(j = 1; j < 8; j++){
24         if(c[j] == 0 && L[j + i] == 0 && R[i - j + n] == 0){
25             c[j] = 1;
26             L[j+i] = 1;
27             R[i - j + n] = 1;
28             if(i < n){
29                 try(i+1);
30             }else{
31                 output();
32             }
33         }
34         c[j] = 0;
35         L[j+i] = 0;
36         R[i - j + n] = 0;
37     }
38 }

```

迭代算法例题 P121

- 兔子繁殖
- 求两个整数的最大公约数
- 猴子吃桃问题
- 杨辉三角
- 穿越沙漠问题
- 求方程式的根
- 牛顿迭代法
- 二分法求解方程

蛮力法 P131

- 百鸡百钱
- 数字谜

分治算法 P136

- 金块问题
- 残缺棋盘
- 求数列的最大子段和
- 大整数乘法
- 选择问题

贪婪算法 P151

- n个数去掉s个新数最少
- 数列极差问题
- 埃及分数
- 币种统计
- 取数游戏

- 求3个数的最大公倍数
- 狱吏问题

动态规划P162

- 数塔问题
- 资源分配问题
- n 个矩阵联乘问题
- 求一个数列最大不下降子序列