

《信息学奥赛一本通·初赛真题解析》

第二章 程序设计基本知识

第1节 程序基本常识

目录

一、算法的特征

二、算法的空间复杂度

三、算法的时间复杂度



算法特征

- **有穷性**：执行有穷步，在有穷的时间内完成。
- **确切性**：每一条指令必须有确切的含义，不会产生歧义。在任何条件下算法只有唯一的一条执行路径。
- **可行性**：算法中的操作可以通过执行有限次来实现。
- **输入**：一个算法有零个或者多个输入。
- **输出**：一个算法有一个或者多个输出。

- **空间复杂度**：指执行算法所需占用的内存空间。
- **时间复杂度**：指算法执行时所需消耗时间，通常用算法执行次数来衡量，记作 $T(n)=O(f(n))$ ，其中 $f(n)$ 是算法执行次数的函数。

时间复杂度举例

- 例1：求出以下算法的时间复杂度。

```
void fundemo(int n){  
    int i=1,j=100;  
    while(i<n){  
        i+=2;  
        ++j;  
    }  
}
```

分析：该算法基本操作为 $i+=2$ 和 $++j$ ，规模为 n 。 i 最后的值为 $1+2m$ (m 代表执行次数)，则 $1+2m+k=n$ (k 是足够小的数)， $m=n-1-k/2$ ，故 $f(n)=n-1-k/2$ ，此时 $f(n)$ 增长最快的那项是 n ，故 $T(n)=O(f(n))=O(n)$ 。



时间复杂度计算规则

- 加法规则

$$T(n,m) = T1(n) + T2(m) = O(\max\{f(n), g(m)\}) \quad // \text{并列算法}$$

- 乘法规则

$$T(n,m) = T1(n) * T2(m) = O(\max\{f(n)*g(m)\}) \quad // \text{嵌套算法}$$

- 比较规则

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < \dots < O(n^K) < O(2^n) < O(n!) < O(n^n)$$



【课堂练习】



1. 【NOIP2011】在使用高级语言编写程序时，一般提到的“空间复杂度”中的“空间”是指()。

- A. 程序运行时理论上所占的内存空间
- B. 程序运行时理论上所占的数组空间
- C. 程序运行时理论上所占的硬盘空间
- D. 程序源文件理论上所占的硬盘空间

2. 【NOIP2013】斐波那契数列的定义如下： $F_1 = 1$ ， $F_2 = 1$ ， $F_n = F_{n-1} + F_{n-2}$ ($n \geq 3$)。如果用下面的函数计算斐波那契数列的第 n 项，则其时间复杂度为()。

```
int F(int n){  
    if (n <= 2)  
        return 1;  
    else  
        return F(n - 1) + F(n - 2);  
}
```

- A. $O(1)$
- B. $O(n)$
- C. $O(n^2)$
- D. $O(Fn)$

3. 【NOIP2013】 $T(n)$ 表示某个算法输入规模为 n 时的运算次数。如果 $T(1)$ 为常数，且有递归式 $T(n) = 2 * T(n / 2) + 2n$ ，那么 $T(n) = ()$ 。

- A. $O(n)$
- B. $O(n \log n)$
- C. $O(n^2)$
- D. $O(n^2 \log n)$

4. 【NOIP2015】设某算法的计算时间表示为递推关系式 $T(n) = T(n - 1) + n$ (n 为正整数) 及 $T(0) = 1$, 则该算法的时间复杂度为()。

A. $O(\log n)$ B. $O(n \log n)$ C. $O(n)$ D. $O(n^2)$

5. 【NOIP2018】设某算法的时间复杂度函数的递推方程是 $T(n) = T(n - 1) + n$ (n 为正整数) 及 $T(0) = 1$, 则该算法的时间复杂度为()。

A. $O(\log n)$ B. $O(n \log n)$ C. $O(n)$ D. $O(n^2)$

6. 【NOIP2016】假设某算法的计算时间表示为递推关系式

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

$$T(1) = 1$$

则算法的时间复杂度为()。

A. $O(n)$ B. $O(\sqrt{n})$ C. $O(\sqrt{n} \log n)$ D. $O(n^2)$

7. 【NOIP2017】若某算法的计算时间表示为递推关系式:

$$T(N) = 2T(N / 2) + N \log N$$

$$T(1) = 1$$

则该算法的时间复杂度为()。

A. $O(N)$ B. $O(N \log N)$ C. $O(N \log^2 N)$ D. $O(N^2)$