

考点 1：时间复杂度与空间复杂度

2009

王道论坛

• 算法题42

2010

王道论坛

• 小题1
• 算法题42

2011

王道论坛

• 小题1
• 小题5
• 算法题42

2012

王道论坛

• 小题1
• 算法题41

2013

王道论坛

• 小题1

2014

王道论坛

• 算法题41

2015

王道论坛

• 算法题43

2016

王道论坛

• 小题1

2017

王道论坛

• 算法题41

2018

王道论坛

• 小题1
• 算法题41

2019

王道论坛

• 算法题41

2020

王道论坛

• 算法题41

2021

王道论坛

• 小题1
• 应用题42

2022

王道论坛

历年考频： 小题×8、算法题×10、应用题×1

1

考点1

时间复杂度
&
空间复杂度

王道考研/CSKAOYAN.COM

2

时间复杂度

【考点笔记】时间复杂度

时间复杂度 $T(n)$ 是指算法中所有语句的频度（执行次数）之和。

人们关心的是当 n 趋于无穷时 $T(n)$ 的数量级，而非 $T(n)$ 的准确大小，因此以 $T(n)$ 的数量级来表征时间复杂度。

数量级最大的一项必定是由最深层循环的语句贡献的，称之为基本运算。

$T(n)$ 与算法中基本运算的频度 $f(n)$ 同数量级，所以通常采用基本运算的频度的数量级 $O(f(n))$ 来分析算法的时间复杂度，记为 $T(n) = O(f(n))$ 。

并列多个循环，
满足加法规则

a) 加法规则

$$T(n) = T_1(n) + T_2(n) = O(f(n)) + O(g(n)) = O(\max(f(n), g(n)))$$

只保留更高阶的项

嵌套多个循环，
满足乘法规则

b) 乘法规则

$$T(n) = T_1(n) \times T_2(n) = O(f(n)) \times O(g(n)) = O(f(n) \times g(n))$$



巴拉拉能量——
数据结构 +20分

常对幂指阶

3

时间复杂度

a) 加法规则

$$T(n) = T_1(n) + T_2(n) = O(f(n)) + O(g(n)) = O(\max(f(n), g(n)))$$

只保留更高阶的项

并列多个循环，
使用加法规则

b) 乘法规则

$$T(n) = T_1(n) \times T_2(n) = O(f(n)) \times O(g(n)) = O(f(n) \times g(n))$$

嵌套多个循环，
使用乘法规则

```
count=0;
for(k=1;k<=n;k*=2)
    count++;
for(j=1;j<=n;j++)
    count++;
```

1. 下列程序段的时间复杂度是_____。

```
count=0;
for(k=1;k<=n;k*=2)
    for(j=1;j<=n;j++)
        count++;
```

A. $O(\log_2 n)$

B. $O(n)$

C. $O(n \log_2 n)$

D. $O(n^2)$

【2014】
嵌套循环

4

时间复杂度

1. 下列函数的时间复杂度是_____。

```
int func(int n){
    int i=0, sum=0;
    while(sum < n) sum += ++i;
    return i;
}
```

- A. $O(\log n)$ B. $O(n^{1/2})$ C. $O(n)$ D. $O(n \log n)$

【2017】循环内的处理影响循环次数

1. 求整数 $n(n \geq 0)$ 阶乘的算法如下，其时间复杂度是_____。

```
int fact(int n){
    if (n <= 1) return 1;
    return n * fact(n-1);
}
```

- A. $O(\log_2 n)$ B. $O(n)$ C. $O(n \log_2 n)$ D. $O(n^2)$

【2012】简单递归

5

空间复杂度

【考点笔记】空间复杂度

空间复杂度 $S(n)$ 指算法运行过程中所使用的**辅助空间的大小**，通常结合算法题考查。

算法原地工作是指算法所需辅助空间是常量，即 $O(1)$ 。

注意影响**辅助空间大小**的两个方面：

- ① 算法运行过程中各种变量所占空间（如：辅助数组）
- ② 递归工作栈带来的空间复杂度（通常和递归深度同等数量级）

注意：有些情况下，算法的时间复杂度、空间复杂度还随问题的输入数据集的不同而不同（最好、最坏、平均）——常结合查找、排序算法考察

6

时间&空间复杂度结合算法题考查

41. (13分) 定义三元组 (a, b, c) (a, b, c 均为整数) 的距离 $D = |a - b| + |b - c| + |c - a|$ 。给定 3 个非空整数集合 S_1, S_2 和 S_3 , 按升序分别存储在 3 个数组中。请设计一个尽可能高效的算法, 计算并输出所有可能的三元组 (a, b, c) ($a \in S_1, b \in S_2, c \in S_3$) 中的最小距离。例如, $S_1 = \{-1, 0, 9\}, S_2 = \{-25, -10, 10, 11\}, S_3 = \{2, 9, 17, 30, 41\}$, 则最小距离为 2, 相应的三元组为 $(9, 10, 9)$ 。要求:

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

```
int findMinDist(int[] S1, int[] S2, int[] S3, int N1, int N2, int N3) {
    int min = INT_MAX;
    for (int i=0; i<N1; i++)
        for (int j=0; j<N2; j++)
            for (int k=0; k<N3; k++)
                //计算 (S1[i], S2[j], S3[k]) 的距离, 并更新 min
                .....
}
```

只传入了数组的指针,
而不是整个数组

Tips: 通常算法题的最后一个问题是分析复杂度, 需要基于自己设计的算法来分析, 即便算法不是最优的, 但复杂度分析正确该小问依然可得满分

时间复杂度= $O(N_1 \cdot N_2 \cdot N_3)$

空间复杂度= $O(N_1 + N_2 + N_3)$ ❌

空间复杂度= $O(1)$ ✅

7

考点 1: 时间复杂度与空间复杂度

【命题重点】

1. 给定代码段, 分析算法的**时间复杂度**。
循环类代码——分析每层循环的循环次数与问题规模 n 之间的关系, 若有多层嵌套循环, 则使用乘法规则
递归类代码——分析**递归次数**与问题规模 n 之间的关系。若要分析**空间复杂度**, 则应找到**递归深度**与问题规模 n 之间的关系。
2. 结合算法题考查, 分析自己设计的算法的时、空复杂度。

8

