

复杂度分析

01 / 计算程序使用时间

做题目这样的要求：

时间限制: C/C++ 1秒, 其他语言2秒
空间限制: C/C++ 262144K, 其他语言524288K
64bit IO Format: %lld

对于空间限制来说, 我们可以通过计算就知道使用的空间大小。
如这里的内存限制是256MB, 即你程序中申请的内存不能超过256MB。如果使用int a[100], 大致需要 $100 * 4 \text{ Byte}$ 。

题目对于空间比较宽松。
一般只要满足以下要求, 就可以过题又不超空间限制。

在函数(非递归)中,
一维数组大小不超过 $1e5$ 。

在全局变量中,
一维数组大小不超过 $1e8$
二维数组大小不超过 $1e4$

数据类型	说明	32位字节数	64位字节数	取值范围
bool	布尔型	1	1	true, false
char	字符型	1	1	-128~127
unsigned char	无符号字符型	1	1	0~255
short	短整型	2	2	-32768~32767
unsigned short	无符号短整型	2	2	0~65535
int	整型	4	4	-2147483648~2147483647
unsigned int	无符号整型	4	4	0~4294967295
long	长整型	4	4	-
unsigned long	无符号长整型	4	4	-
long long	长整型	8	8	$-2^{64} \sim 2^{64}-1$
float	单精度浮点数	4	4	范围: $-2^{128} \sim 2^{128}$ 精度为6~7位有效数字
double	双精度浮点数	8	8	范围: $-2^{1024} \sim 2^{1024}$ 精度为15~16位
long double	扩展精度浮点数	8	8	范围: $-2^{1024} \sim 2^{1024}$ 精度为15~16位

8bit(位)=1Byte(字节)

1024Byte(字节)=1KB

1024KB=1MB

1024MB=1GB

1024GB=1TB

02 / 算法复杂度分析

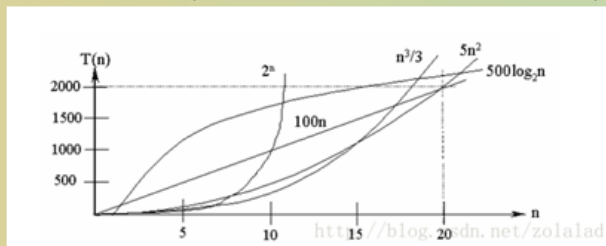
常见的算法时间复杂度由小到大依次为：

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

其中 $O(\log_2 n)$ 、 $O(n)$ 、 $O(n \log_2 n)$ 、 $O(n^2)$ 和 $O(n^3)$ 称为多项式时间 而 $O(2^n)$ 和 $O(n!)$ 称为指数时间

计算机科学家普遍认为前者（即多项式时间复杂度的算法）是有效算法, 把这类问题称为 **P (Polynomial, 多项式) 类问题**, 而把后者（即指数时间复杂度的算法）称为 **NP (Non-Deterministic Polynomial, 非确定多项式) 问题**。

在一般的问题中, 多项式复杂度的算法是可以接受的算法, 而指数时间复杂度的算法则认为效率太低,



02 / 算法复杂度分析

问题规模n 与 可用算法的复杂度

问题规模n	可用算法的时间复杂度					
	$O(\log_2 n)$	$O(n)$	$O(n \log_2 n)$	$O(n^2)$	$O(2^n)$	$O(n!)$
$n < 12$	√	√	√	√	√	√
$n < 26$	√	√	√	√	√	×
$n \leq 5e3$	√	√	√	√	×	×
$n \leq 1e6$	√	√	√	×	×	×
$n \leq 1e7$	√	√	×	×	×	×
$n \leq 1e8$	√	×	×	×	×	×

进制转化

十进制：13

二进制：1101

$$13 = 1 * 2^0 + 0 * 2^1 + 1 * 2^2 + 1 * 2^3$$