黑猫编程 blackcat1995.com 算法定义

- 算法（Algorithm）是对特定问题求解步骤的一种描述，是指令的有限序列，其中，每条指令表示一个或多个操作，具有下列五个重要特性：
  ① 有穷性：一个算法必须总在执行有穷步之后结束，且每一步在又穷时间内完成。
  ② 确定性：算法中每条指令必须有确切的含义，对于相同的输入只能得出相同的输出。
  ③ 可行性：算法中描述的操作都可以通过已经实现的基本运算执行有限次来实现。
  ④ 输入：一个算法有零个或多个输入，这些输入取自于某个特定对象的集合。
  ⑤ 输出：一个算法有一个或多个输出，这些输出是与输入有着某种特定关系的量。

- 一个好的算法应考虑达到以下目标：正确性、可读性、健壮性、效率与低存储需求。

黑猫编程 blackcat1995.com 时间复杂度

- 一个语句的频度是指该语句在算法中被重复执行的次数，算法中所有语句的频度之和记为T(n)，是该算法问题规模为n的函数，时间复杂度主要分析T(n)的数量级。
- 时间复杂度记为 T(n) = O(f(n))
- O 的含义是T(n)的数量级，严格的数学定义是：若T(n)和f(n)是定义在正整数集合上的两个函数，若存在正常数C和n_0，使得当n≥ n_0时，都满足0≤T(n) ≤Cf(n)。
- 加法原则：T(n)=$T_1(n) + T_2(n)$=O(f(n))+O(g(n))=O(max(f(n),g(n)))
- 乘法原则：T(n)=$T_1(n) \times T_2(n)$=O(f(n)) × O(g(n))=O(f(n) × g(n))
- 常见渐进时间复杂度：O(1)<O(logn)<O(nlogn)<O(n^2)<O(n^3)<O(2^n)<O(n!)<O(n^n)

```
void f(int n){
    int i = 1;
    while(i <= n)
        i *= 2;
}
```

时间复杂度：$O(\log_2 n)$

```
x = 0
while(n >= (x + 1) * (x + 1))
    x++;
```

时间复杂度：$O(\sqrt{n})$

黑猫编程
blackcat1995.com

- 算法的空间复杂度S(n)是指该算法所耗费的存储空间，是问题规模n的函数，记为S(n)=O(g(n))。
- 一个程序在执行时需要存储空间来存放本身所用的指令、常数、变量和输入数据外，还需要一些对数据进行操作的工作单元和存储一些为实现计算所需信息的存储空间。若输入数据所占空间只取决于问题本身，和算法无关，则只需分析除输入和程序之外的额外空间。
- 算法原地工作是指算法所需的辅助空间为常量，即O(1)。