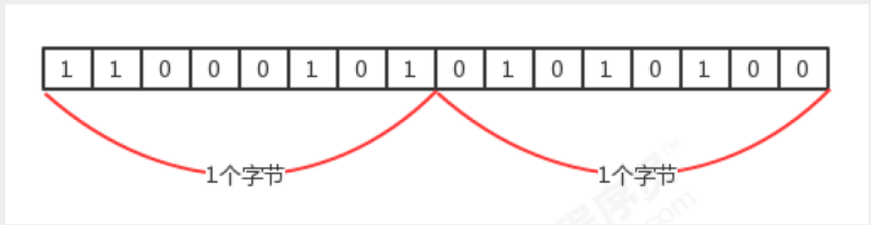
算法分析

时间复杂度分析

- 事后分析估算方法
  - 这种统计方法主要是通过设计好的测试程序和测试数据，利用计算机计时器对不同的算法编制的程序的运行时间进行比较，从而确定算法效率的高低，但是这种方法有很大的缺陷：必须依据算法实现编制好的测试程序，通常要花费大量时间和精力，测试完了如果发现测试的是非常糟糕的算法，那么之前所做的事情就全部白费了，并且不同的测试环境(硬件环境)的差别导致测试的结果差异也很大。
- 事前分析估算方法
  - 影响因素
    - 算法采用的策略和方案
    - 编译产生的代码质量
    - 问题的输入规模
    - 机器执行指令的速度
  - 函数渐近增长
    - 1. 算法函数中的常数可以忽略；
    - 2.算法函数中最高次幂的常数因子可以忽略；
    - 3.算法函数中最高次幂越小，算法效率越高。
  - 时间复杂度
    - 大O记法 ＝
      - 1.用常数1取代运行时间中的所有加法常数；
      - 2.在修改后的运行次数中，只保留高阶项；
      - 3.如果最高阶项存在，且常数因子不为1，则去除与这个项相乘的常数；
    - 常见的大O阶
      - 1.常数阶 — 普通语句 — O(1)
      - 2.对数阶 — 二分策略 — O(logN)
      - 3.线性阶 — 单层循环 — O(N)
      - 4.线性对数 — 分治思想 — O(NlogN)
      - 5.平方阶 — 双层循环 — O(N^2)
      - 6.立方阶 — 三层循环 — O(N^3)
      - 7.指数阶 — 穷举查找 — O(2^N)

空间复杂度分析

- 常见内存占用
  - 数据访问方式



一次一个字节

  - 基本数据类型

基本数据类型内存占用情况

	数据类型	byte	short	int	long	float	double	boolean	char
	内存占用字节数	1	2	4	8	4	8	1	2

  - 引用数据类型
    - 一个引用（机器地址）需要8个字节表示
  - 创建一个对象
    - 比如new Date()，除了Date对象内部存储的数据(例如年月日等信息)占用的内存，该对象本身也有内存开销，每个对象的自身开销是16个字节，用来保存对象的头信息。
  - 内存自动填充
    - 一般内存的使用，如果不够8个字节，都会被自动填充为8字节
- 空间复杂度 — 计算标记公式
  - S(n)=O(f(n)),其中n为输入规模，f(n)为语句关于n所占存储空间的函数。