

## 第五讲： 循环、递归、静态方法（子函数）、数组

1. 递归:  $n! = (n-1)! * n$   $0! = 1! = 1$ 
  - a) 递归基础(出口) `//0!=1!=1`
  - b) 递归体 `//f(n)=f(n-1)*n`
  - c) 数学归纳法的思想 (vs 递推)
  - d) 算法:
    - i. 阶乘  $n!$ 
      1.  $N \leq 12$  int
      2.  $N \leq 20$  long
      3. BigInteger 、 数学方法
    - ii. 组合数:
      1. 基于阶乘的计算  $C(n,r) = n! / (r! * (n-r)!)$  //计算范围小  $n, r \leq 20$
      2. 基于递归函数  $C(n,r) = C(n-1,r-1) + C(n-1,r)$  //双递归很慢 ,
      3. 基于数组优化 (杨辉三角)
    - iii. 最大公约数 (最小公倍数 LCM) //Greatest Common Divisor
      1. `If(b==0) return a`
      2. `Return gcd(b,a%b)`
      3. `Gcd(a,b)*lcm(a,b)=a*b;`
2. 一维数组: `arr.length;`
  - a) 定义:
    - i. `int a[]; int[] b,c; //null`
    - ii. `int [] d={1,2,3};//d.length=3,d[0]=1,d[1]=2,d[2]=3`
    - iii. `int [] e=new int[3];//e.length=3,e[0]=e[1]=e[2]=3`
  - b) 输出:
    - i. 索引输出 (读写)
    - ii. `For_each` (只读)
    - iii. `Arrays.toString(arr);`//调试
  - c) 数组的拷贝:
    - i. 浅拷贝: `b=a`
    - ii. 深拷贝/克隆
      1. `B=new int[a.length];for(int i=0;i<a.length;i++)B[i]=a[i];`
      2. `B=a.clone()`
  - d) 数组应用:
    - i. 递推
    - ii. 计算最大、最小、求和
    - iii. 填充 `Arrays.fill`、排序 `Arrays.sort` 和查找 `Arrays.binarySearch`
      1. 简单排序 (演示)  $O(n^2)$ 
        - a) 选择排序
        - b) 冒泡排序
      2. 高级排序:  $O(n \log n)$ 
        - a) 快速排序
        - b) 归并排序

c) 堆排序

iv. 素数筛法: 哥德巴赫猜想的验证

1. 把  $0 \sim N-1$  都当成素数候选
2. 删除 0 和 1
3. 从  $a=2$  to  $\sqrt{N}$  中的每一个数,  
如果素数  $a$ , 保留  $a$  删除  $a$  的所有整数倍

3. 二维数组

- a) 二维数组的定义 (一维数组, 每一个元素都是一维数组)
- b) 二维数组的应用 :
  - i. 杨辉三角
  - ii. 幻方