



大学生计算与信息化素养

算法策略：穷举与递归

内 容



```
graph LR; A((内容)) --- B[算法策略概述]; A --- C[穷举]; A --- D[递归];
```

算法策略概述

穷举

递归

穷举

递归

分治

动态
规划

贪心

回溯

引例：



引例：暴力破解密码

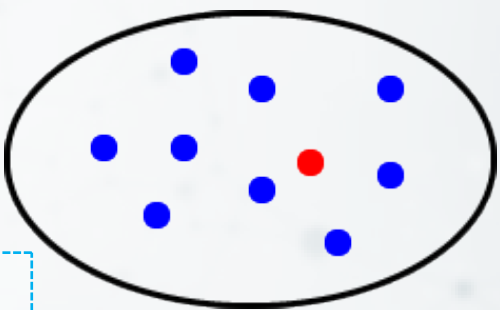
简单来说就是将密码进行逐个推算直到找出真正的密码为止。

- (1) 4位，数字，共有10000种组合
- (2) 8位，有大小写字母、数字、符号，共有几千万亿种组合



概念：

对问题所有的可能的解逐一尝试，从而找出问题的真正解。

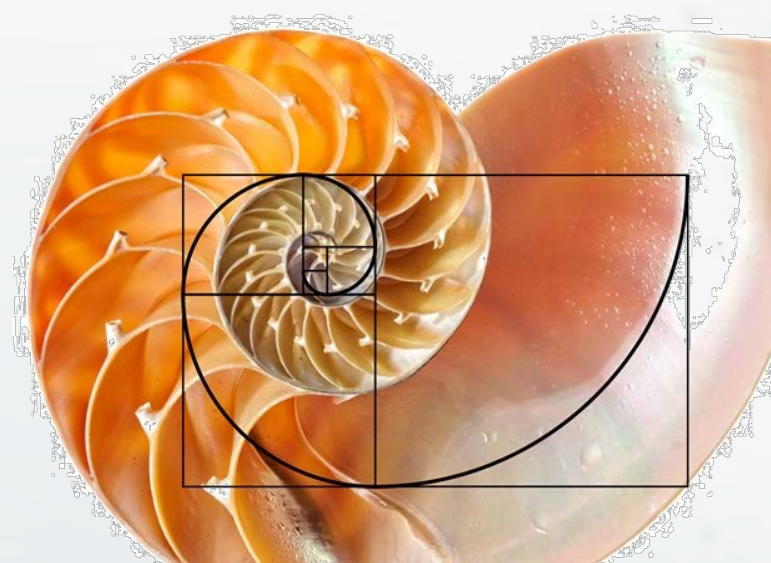
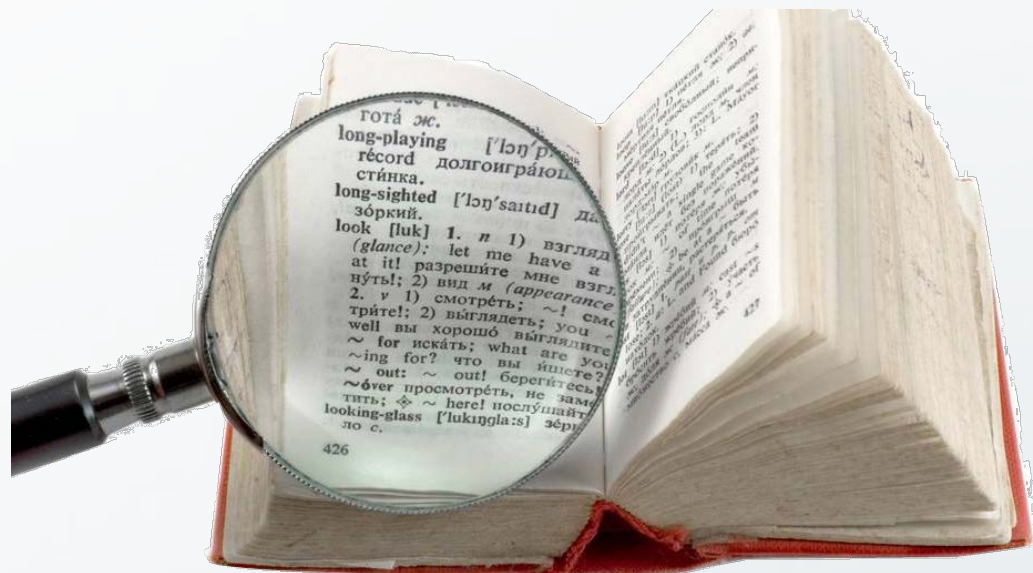


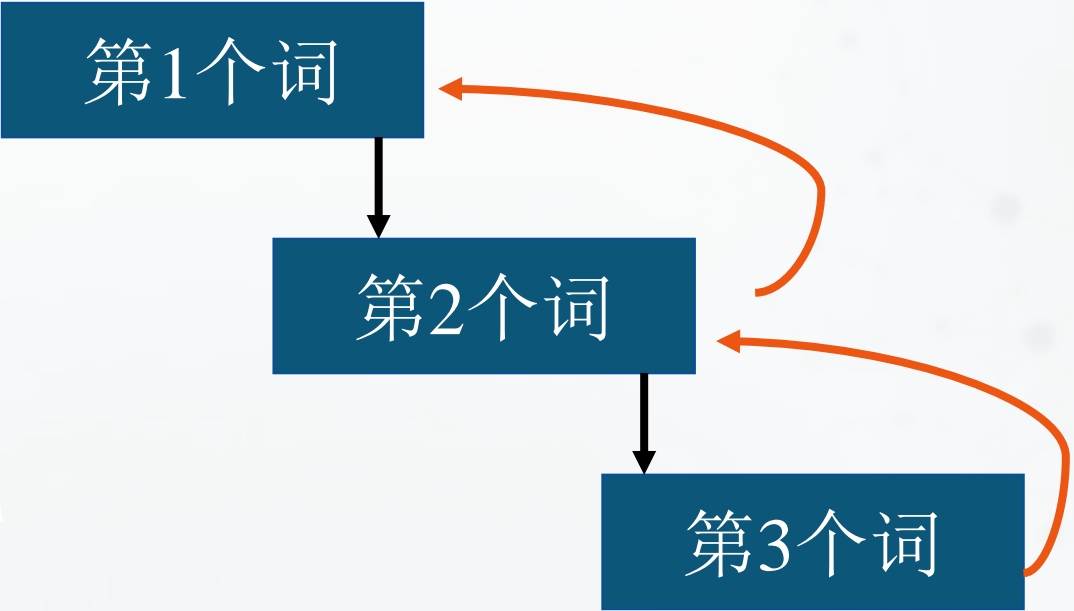
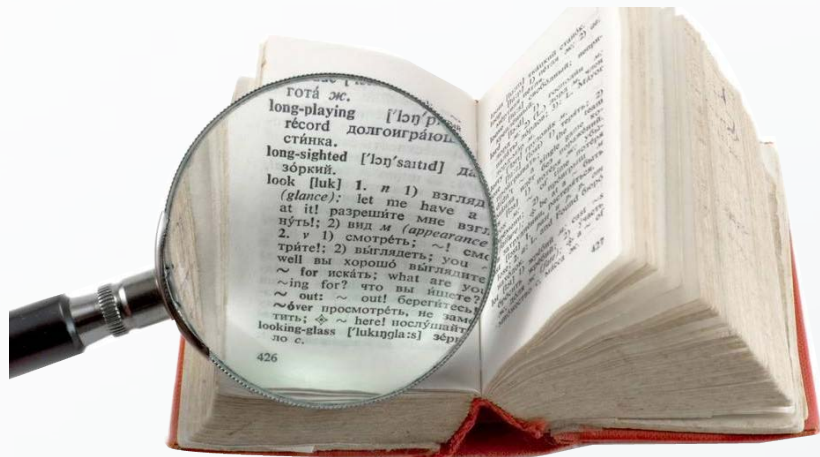
- 优点：
 - 简单容易实现、可以得到真正解
- 缺点：
 - 效率低
- 适用：
 - 解空间有限，其他策略不容易实现

递归



引例:





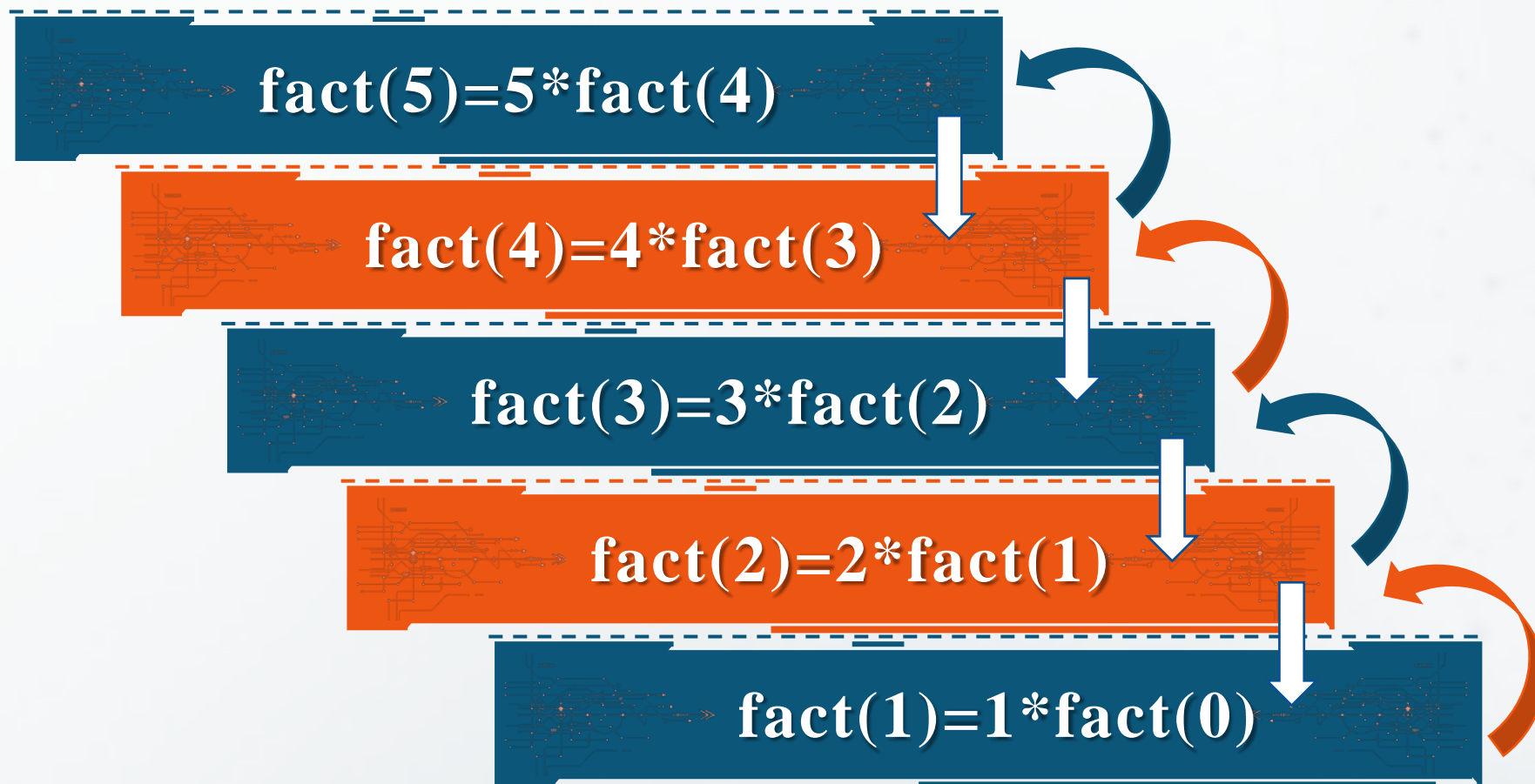
递归



求n的阶乘

$\text{fact}(n)$

$\text{fact}(n) = n * \text{fact}(n-1), \text{fact}(0) = 1$





递归



概念：为求解规模为 n 的问题，通过重复将问题分解为同类的子问题而解决问题的方法。特别的，当规模 $n = 1$ 或 0 时，能直接得解，即递归出口

算法描述：

1. 分解问题，将大问题规模缩小，子问题变成与大问题形式相同，找到子问题与大问题之间关系，形成递归表达式

$$\text{fact}(n)=n*\text{fact}(n-1)$$

2. 给出递归出口

$$\text{fact}(0)=1$$

小 结



```
graph LR; A((小结)) --- B[算法策略概述]; A --- C[穷举]; A --- D[递归];
```

算法策略概述

穷举

递归