

# 算法入门系列课程1 - 周而复始

## 算法概述

### 1. 什么是算法？

解决问题的正确方法和具体的实施步骤。

例子1：如何在两栋相距50m的大楼的两个房间牵一条线（两个房间都有窗）？

- 养一只鸟（如鸽子），将线送过去
- 用很长的杆子将线递过去
- 用无人机（遥控飞行器）将线送过去

如何评价这些方法的好坏？ **少花钱，不费事！**

例子2：大教室里坐了几百名学生一起听课，如何快速的统计学生人数？

例子3：向列表容器中**逆向**插入100000个元素。

- 方法1：

```
nums = []
for i in range(100000):
    nums.append(i)
nums.reverse()
```

- 方法2：

```
nums = []
for i in range(100000):
    nums.insert(0, i)
```

例子3：生成Fibonacci数列（前100个Fibonacci数）。

- 方法1 - 递推：

```
a, b = 0, 1
for num in range(1, 101):
    a, b = b, a + b
    print(f'{num}: {a}')
```

- 方法2 - 递归：

```
def fib(num):
    if num in (1, 2):
        return 1
    return fib(num - 1) + fib(num - 2)

for num in range(1, 101):
    print(f'{num}: {fib(num)}')
```

- 方法3 - 改进的递归：

```
def fib(num, temp={}):
    if num in (1, 2):
        return 1
    elif num not in temp:
        temp[num] = fib(num - 1) + fib(num - 2)
    return temp[num]
```

- 方法4 - 改进的递归:

```
from functools import lru_cache

@lru_cache()
def fib(num):
    if num in (1, 2):
        return 1
    return fib(num - 1) + fib(num - 2)
```

## 2. 如何评价算法的好坏?

[渐近时间复杂度](#)和渐近空间复杂度。

## 3. 大O符号的意义?

表示一个函数相对于输入规模的增长速度，也可以称之为函数的数量级。

大O符号	说明	例子
$O(1)$	常量时间复杂度	布隆过滤器 / 哈希存储
$O(\log_2 n)$	对数时间复杂度	二分查找 (折半查找)
$O(n)$	线性时间复杂度	顺序查找 / 桶排序
$O(n \cdot \log_2 n)$	对数线性时间复杂度	高级排序算法 (归并排序、快速排序)
$O(n^2)$	平方时间复杂度	简单排序算法 (选择排序、插入排序、冒泡排序)
$O(n^3)$	立方时间复杂度	Floyd算法 / 矩阵乘法运算
$O(2^n)$	几何级数时间复杂度	汉诺塔
$O(n!)$	阶乘时间复杂度	旅行经销商问题

## 穷举法

在计算机科学中，**穷举法**或者**暴力搜索法**是一个非常非常直观的解决问题的方法，这种方法通过一项一项的列举解决方案所有可能的候选项以及检查每个候选项是否符合问题的描述，最终得到问题的解。

虽然暴力搜索很容易实现，并且如果解决方案存在它就一定能够找到，但是它的代价是和候选方案的数量成比例的，由于这一点，在很多实际问题中，消耗的代价会随着问题规模的增加而快速地增长。因此，当问题规模有限或当存在可用于将候选解决方案的集合减少到可管理大小时，就可以使用暴力搜索。另外，当实现方法的简单度比速度更重要时，也可以考虑使用这种方法。

## 经典例子

1. **百钱百鸡**问题：公鸡5元一只，母鸡3元一只，小鸡1元三只，用100元买一百只鸡，问公鸡、母鸡、小鸡各有多少只？

```
for x in range(21):
    for y in range(34):
        z = 100 - x - y
        if z % 3 == 0 and 5 * x + 3 * y + z // 3 == 100:
            print(x, y, z)
```

2. **五人分鱼**问题：ABCDE五人在某天夜里合伙捕鱼，最后疲惫不堪各自睡觉。第二天A第一个醒来，他将鱼分为5份，扔掉多余的1条，拿走了属于自己的一份；B第二个醒来，也将鱼分为5份，扔掉多余的1条，拿走属于自己的一份；然后C、D、E依次醒来，也按同样的方式分鱼，问他们至少捕了多少条鱼？

```
fish = 6
while True:
    total = fish
    enough = True
    for _ in range(5):
        if (total - 1) % 5 == 0:
            total = (total - 1) // 5 * 4
        else:
            enough = False
            break
    if enough:
        print(fish)
        break
    fish += 5
```

3. **暴力破解口令**：

```
import re

import PyPDF2

with open('Python_Tricks_encrypted.pdf', 'rb') as pdf_file_stream:
    reader = PyPDF2.PdfFileReader(pdf_file_stream)
    with open('dictionary.txt', 'r') as txt_file_stream:
        file_iter = iter(lambda: txt_file_stream.readline(), '')
        for word in file_iter:
            word = re.sub(r'\s', '', word)
            if reader.decrypt(word):
                print(word)
                break
```