



问题解答

两周内容小结: W05/06: 递归

关于H4作业

# 数据结构与算法 ( Python ) -10/0327

陈斌 gischen@pku.edu.cn 北京大学地球与空间科学学院

# 目录

- › 问题解答
- › 两周内容小结  
W05/06: 递归
- › 关于H4作业



# 问题解答？

- › 请大家发在Canvas讨论里。
- › 随后解答

# 本章目标

- › 了解某些难解的问题具有简单的递归解决法；
- › 学会如何用递归方式写程序；
- › 了解和应用递归的“三定律”；
- › 了解递归是迭代iteration的一种形式；
- › 实现问题的递归描述；
- › 了解递归在计算机系统中是如何实现的。

# W05/06 : 递归

- › 401 什么是递归 13m20s
- › 402 递归的应用：任意进制转换 6m32s
- › 405 递归可视化：谢尔宾斯基三角 10m31s
- › 403 递归调用的实现 8m32s
- › 404 递归可视化：分形树 10m54s
- › 406 递归的应用：汉诺塔 10m37s
- › 407 递归的应用：探索迷宫 16m35s

# 递归 “三定律”

❖ 为了向阿西莫夫的“机器人三定律”致敬，递归算法也总结出“三定律”

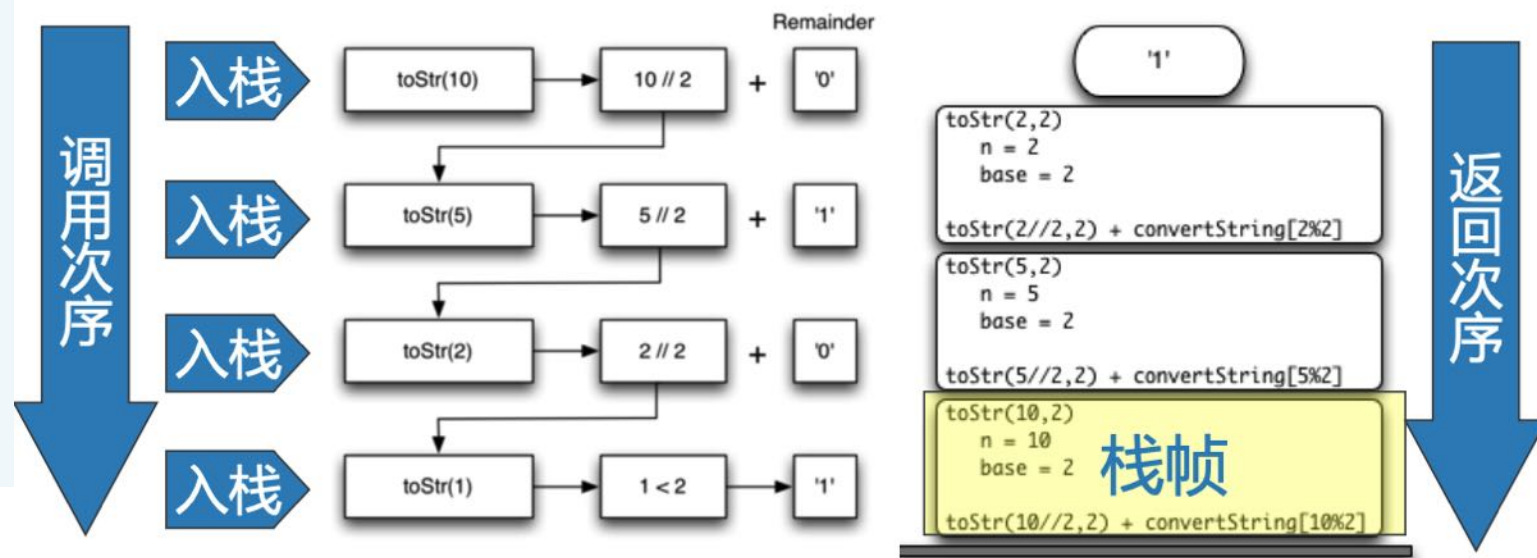
- 1, 递归算法必须有一个基本结束条件（最小规模问题的直接解决）
- 2, 递归算法必须能改变状态向基本结束条件演进（减小问题规模）
- 3, 递归算法必须调用自身（解决减小了规模的不同问题）

# 递归调用的实现

❖ 当一个函数被调用的时候，系统会把调用时的**现场数据**压入到**系统调用栈**

每次调用，压入栈的现场数据称为**栈帧**

当函数返回时，要从调用栈的栈顶取得返回地址，恢复现场，弹出栈帧，按地址返回。



# W05/06 : 递归

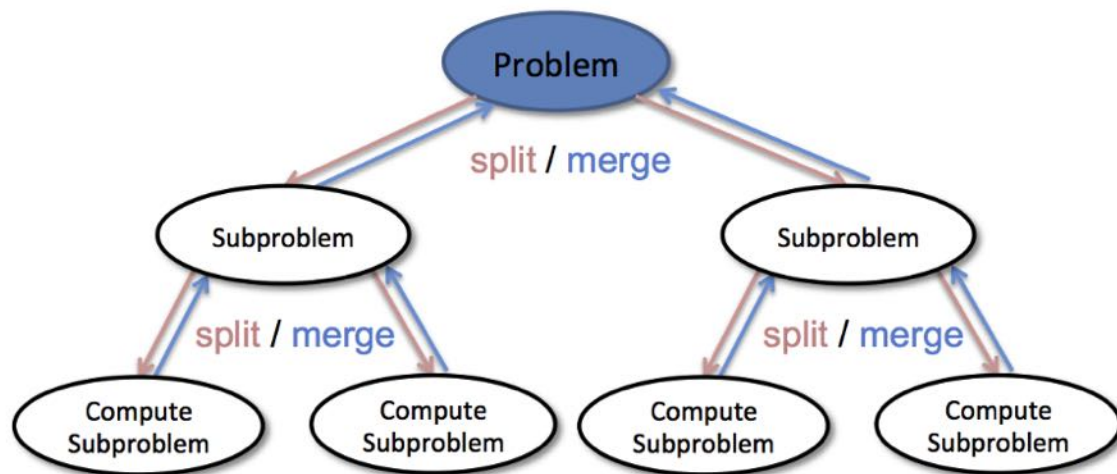
- › 408 分治策略 5m34s
- › 409 优化问题和贪心策略 7m49s
- › 410 找零兑换问题的递归解法 12m32s
- › 411 找零兑换问题的动态规划解法 13m26s
- › 412 动态规划案例分析 17m53s
- › 413 递归小结 6m51s

# 分治策略

## ❖ 解决问题的典型策略：分而治之

将问题分为若干更小规模的部分

通过解决每一个小规模部分问题，并将结果汇总  
得到原问题的解



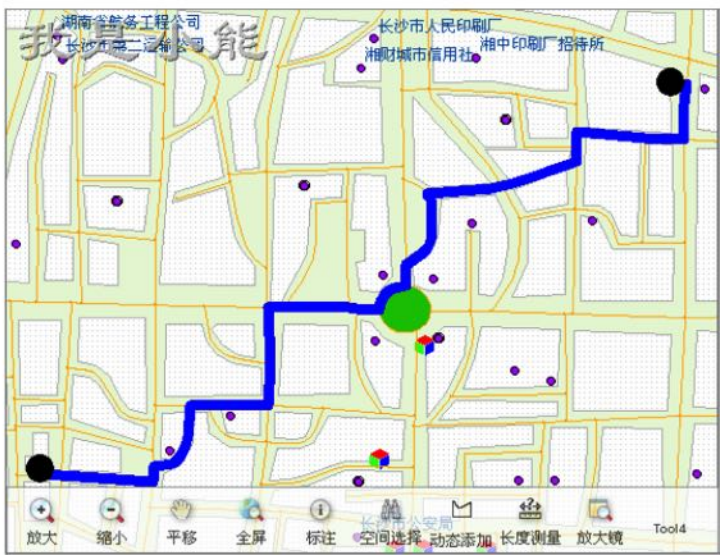
# 优化问题

## ❖ 计算机科学中许多算法都是为了找到某些问题的最优解

例如，两个点之间的最短路径；

能最好匹配一系列点的直线；

或者满足一定条件的最小集合



# 贪心策略Greedy Method

## ❖ 贪心策略

因为我们每次都试图解决问题的**尽量大**的一部分  
对应到兑换硬币问题，就是每次以**最多数量**的**最大面值**硬币来**迅速减少**找零面值

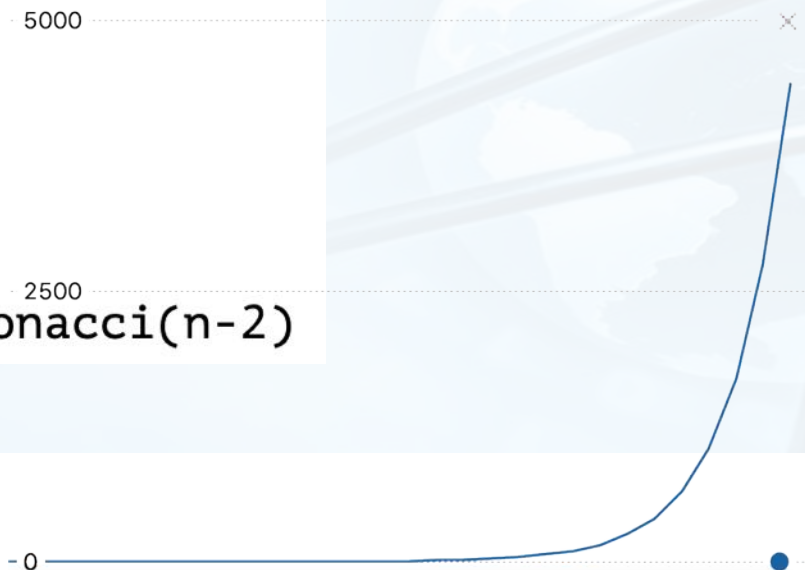
## ❖ “贪心策略” 解决找零兑换问题，在美元或其他货币的硬币体系下表现尚好



# 递归的问题

- › 用分治策略来设计算法，用递归最直接  
算法简洁易懂
- › 但递归方法的问题在于重复计算，导致计算量爆炸  
你猜猜用这个能计算到斐波那契数列的第几个？

```
1 def fibonacci(n):  
2     if n == 1:  
3         return 0  
4     elif n == 2:  
5         return 1  
6     else:  
7         return fibonacci(n-1) + fibonacci(n-2)
```



# 函数值缓存/备忘录技术解决递归重复计算

7500 X

5000

2500

-0

```
1 fib = [-1]*100
2
3 def fibonacci(n):
4     if n == 1:
5         return 0
6     elif n == 2:
7         return 1
8     elif fib[n] >= 0:
9         return fib[n]
10    else:
11        fib[n-1]= fibonacci(n-1)
12        fib[n-2]= fibonacci(n-2)
13        return fib[n-1] + fib[n-2]
```

# 从函数值缓存到动态规划

## 找零兑换：动态规划算法

- ❖ 采用动态规划来解决11分钱的兑换问题  
从1分钱兑换开始，逐步建立一个兑换表

Change to Make

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|----|----|
| 1 |   |   |   |   |   |   |   |   |   |    |    |
| 2 | 1 |   |   |   |   |   |   |   |   |    |    |
| 3 | 1 | 2 |   |   |   |   |   |   |   |    |    |
| 4 | 1 | 2 | 3 |   |   |   |   |   |   |    |    |
| 5 | 1 | 2 | 3 | 4 |   |   |   |   |   |    |    |
| 6 | 1 | 2 | 3 | 4 | 1 |   |   |   |   |    |    |

...

| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 1 |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 1 | 2 |

# 八皇后问题

```
1  def queen(A, cur=0):
2      if cur == len(A):
3          print(A)
4          return 0
5      for col in range(len(A)):
6          A[cur], flag = col, True
7          for row in range(cur):
8              if A[row] == col or abs(col - A[row]) == cur - row:
9                  flag = False
10                 break
11         if flag:
12             queen(A, cur+1)
13 queen([None]*8)
```

# 关于本周作业

- › **必做：**见Canvas页面和gis4g公告  
慕课在线测验，和【H4】动态规划作业
- › **选做：**慕课的OJ作业
- › **【H4】作业提交PyLn平台（见gis4g网站）**

# 【H4】动态规划作业

- › H4-1 博物馆大盗
- › H4-2 单词最短编辑距离

- › **参考资料：**

视频412 动态规划案例分析

<http://gis4g.pku.edu.cn/course/pythonds/#DDL>

- › **DDL 4月1日 18点**