

算法作业 - 02

《2-5 整数划分》

A. 问题描述

将正整数 n 表示为一系列正整数之和,

$$n = n_1 + n_2 + \dots + n_k \text{ (其中, } n_1 \geq n_2 \geq \dots \geq n_k \geq 1, k \geq 1 \text{)}$$

正整数 n 的这种表示称为正整数 n 的划分。正整数 n 的不同的划分的个数称为正整数 n 的划分数, 记作 $p(n)$ 。

例如, 正整数 6 有如下 11 种不同的划分, 所以 $p(6)=11$ 。

6;
5+1;
4+2, 4+1+1;
3+3, 3+2+1, 3+1+1+1;
2+2+2, 2+2+1+1, 2+1+1+1+1;
1+1+1+1+1+1;

B1. 问题分析

求正整数 n 的划分可以转化成求所有最大元素 m ($1 \leq m \leq n$) 加上其 m 对应的正整数 $n-m$ 的划分之和。

B2. 递归式分析

构建一个全局数组 `array[MAX]`, 数组的每一个单元用于按顺序存放当前划分的具体划分组成元素。

设置一个全局变量 `amount`, 用于记录总的划分数。

构建得到最大加数不大于 `max` 的划分记作函数 `division(max, index)`, `index` 表示当前划分中调用 `division` 函数能得到的 `array` 数组的第 `index` 个元素。

可以得到:

(1). 若 `division(0, index)`, `index` ≥ 1

则表示当前划分已经得到, 具体划分为保存在 `array` 数组中的前 `index` 个元素, 然后调用输出函数将该具体划分重定位到标准输出上, 并且每输出一个元素, 便使 `amount` 自增 1。

(2). 若 `division(max, index)`, `max` $\neq 0$, `index` ≥ 1

则设置变量 `i` 从 `max` 迭代到 1, `i` 代表不大于最大加数 `max` 的正整数

1. 若 `index == 0`,

即要得到划分中的第一个元素 (而且是每一种划分中的最大元素)。很明显这第一个元素可以是小于等于 `max` 的任意正整数。然后设置 `array[index]` 为变量 `i` 并且递归调用 `division[max-i, index+1]`, 为了得到最大加数为 `max-i` 的划分, 并且在就近一层递归调用中得到 `array[index+1]` 的值。

2. 若 `i > array[index-1]`,

显然这是不符合题设要求“ $n_1 \geq n_2 \geq \dots \geq n_k \geq 1$ ”, 所以得增加一个约束条件“ $i \leq \text{array}[\text{index}-1]$ ”, 然后才能设置 `array[index]` 为变量 `i` 并且递归调用 `division[max-i, index+1]`, 为了得到最大加数为 `max-i` 的划分, 并且在就近一层递归调用中得到 `array[index+1]` 的值。

B3. 算法时间复杂度分析

老师，我这算法的时间复杂度我花了很久都没算出来，老师能不能帮我算一下，谢谢。

C 算法描述

```
division(max, index)
    if max == 0
        output(index)           //输出当前划分的具体元素
    else
        for i = max to 1
            if index == 0 or i <= array[index-1]
                array[index] = i
                division(max-i, index+1)    //递归调用 division 函数
```

D. 程序（我分别用 C++ 和 Python 都实现了一遍）

!!!C++代码：division.cpp

```
#include <stdio.h>
#include <iostream>

#define MAX 100

using namespace std;

int array[MAX] = {0};

static int amount = 0;

void output(int);

void division(int, int);

int
main() {
    int input;

    cout<<"Please enter a nonnegative integer: ";
    cin>>input;

    division(input, 0);
```

```

        cout<<endl<<"\t"<<amount<<" ways of division"<<endl<<endl;

        return 0;
    }

void
output(int num) {
    int i;
    for (i = 0; i < num; i++)
        cout<<array[i]<<" ";
    cout<<endl;

    amount++;
}

void
division(int max, int index) {
    int i;
    if (max == 0)
        output(index);
    else
        for(i = max; i > 0; i--) {
            if (index == 0 || i <= array[index - 1]) {
                array[index] = i;
                division(max - i, index + 1);
            }
        }
}

!!! Python 代码: division.py
#!/usr/bin/python
# Filename: division.py

array = [0 for i in range(0, 100)]
amount = 0

def Output(num):
    global amount
    for i in range(0, num):
        print array[i],
    amount += 1
    print

def Division(Max, Index):

```

```
if Max == 0:
    Output(Index)
else:
    for i in range(Max, 0, -1):
        if Index == 0 or i <= array[Index - 1]:
            array[Index] = i
            Division(Max - i, Index + 1)

Input = int(raw_input("Please enter a nonnegative integer: "))
Division(Input, 0)
print '\t' + str(amount) + ' ways of division\n'
```

虞嘉豪

学号：0304110518