

# 暨南大学本科实验报告专用纸

课程名称 算法分析与设计实验 成绩评定             
实验项目名称 最少硬币问题 指导教师 李展  
实验项目编号 实验七 实验项目类型 综合性 实验地点             
学生姓名 张印祺 学号 2018051948  
学院 信息科学技术 系 计算机科学 专业 网络工程  
实验时间 2020 年 5 月 6 日

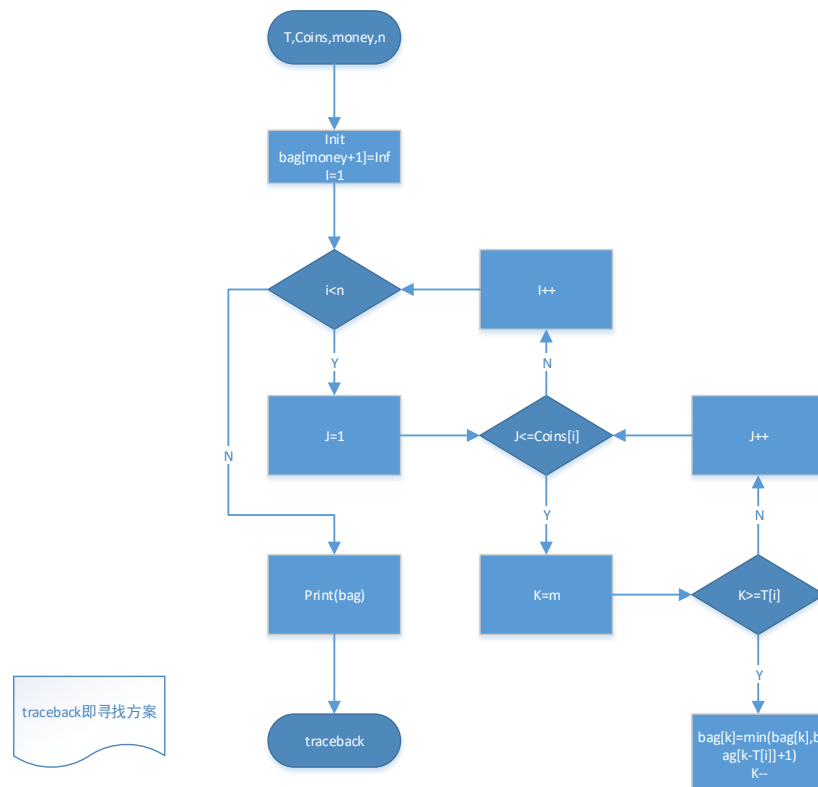
## 一、问题描述

设有  $n$  种不同面值的硬币，各硬币的面值存在于数组  $T[1:n]$  中。现要用这些面值的硬币来找钱。可以使用的各种面值的硬币个数存于数组  $Coins[1:n]$  中。对任意钱数  $0 \leq m \leq 20001$ ，设计一个最少硬币找钱  $m$  的方法

## 二、算法思路

第一个循环：使得可以每一种钱币进行各自分析  
第二个循环：用该钱币数量进行控制循环  
第三个循环：状态迁移控制进行列表填充

## 三、流程图



## 四、测试结果

```

> M4
Coins=[0,2,3,3]
T=18: int
m=18
dpcharge(T,Coins,m)
共需要 5 枚硬币
1 元 1 张
2 元 1 张
5 元 3 张
<__main__.dpcharge at 0x20c2eba83c8>

> M4
Coins=[0,9,15,4]
T=[0,1,2,5]
m=43
dpcharge(T,Coins,m)
共需要 16 枚硬币
1 元 1 张
2 元 11 张
5 元 4 张
<__main__.dpcharge at 0x20c2eba82b0>
```

## 五、实验总结

本算法使用了一维数组做备忘录  
迁移方程:  $m(i,j)=\min\{m(i-1,j),m(i,j-v_i)+1\}$   
时间复杂度为  $O(n^2 \times m)$   
空间复杂度  $T(m)$   
要算出物品的选择, 需要调用 traceBack 算法

## 六、附录 (程序代码)

```
class dpcharge:
    def __init__(self,T:list,Coins:list,money:int):
        n=len(Coins)
        bag=[999999 for i in range(money+1)]
        bag[0]=0
        for i in range(1,n):
            for j in range(1,Coins[i]+1):
                for k in range(m,T[i]-1,-1):
                    bag[k]=min(bag[k],bag[k-T[i]]+1)
        if bag[m]<m:
            print('共需要',bag[m], '枚硬币')
            self.tarceback(T,Coins,m,n,bag[m])
        else:print("Can't charge!")
    def tarceback(self,T,Coins,m,n,x):
        temp=[0 for i in range(n)]
        left=m
```

```
for i in range(n-1,0,-1):
    while temp[i]<Coins[i] and left>=T[i] and sum(temp)<x:
        temp[i]+=1
        left-=T[i]
for i in range(1,n):
    print(T[i], '元', temp[i], '张')
```