算法设计与分析文档

根据网站提供的数据，数据比较多，调试代码的时候如果一个一个的输入的话比较费时间，所以我把数据放在一个EXCEL表格里，这样代码虽然比输入的时候要复杂一些，但是可以在调试的时候方便一点。

利用Python语言进行调试：

前期处理的流程大致如下



导入调试所必要的数据包：xlrd、numpy、pandas等

首先利用xlrd数据包对excel中的数据进行读取： data= xrld.open\_excel(address)

将读取的数据按不同类型提取出来，方便后面进行数据操作

将数据不同的属性，将数据分为： 价格数据、其它条件数据、约束数据

用numpy数据包中的append方法对数据进行操作，将不同属性的数据分别归到不同的数组中：

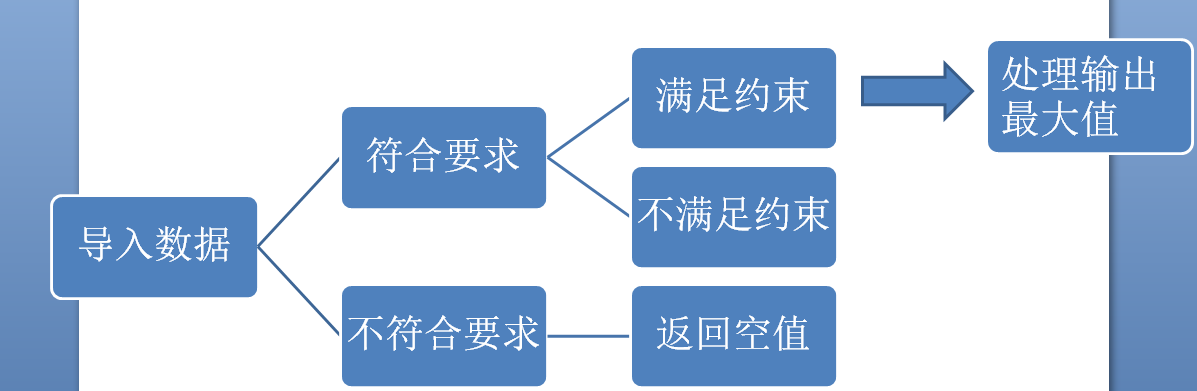
Data = data.Append（条件）

将不同属性的数据归纳到不同的数组之后，进行数据格式的转化，将数组格式转化为矩阵的格式，这里用到pandas数据包：

Data = pd.dataframe(data)

算法实现过程：

大致流程：



用DEF函数定义方法，用来实现算法

在定义的方法内对前面数据处理的时候的变量进行申明（Python中自定义函数不可以调用函数外的变量，如果要调用，必须进行申明）：

Global name,name,name…….

对数据进行筛选，如果数据符合要求，则执行下一步的程序，如果不符合要求，则返回空值。

If 放入背包中的物品数量大于物品的总数

或者放入背包中物品的价值大于物品的总价值

数据则不符合要求，返回空值（return 0）

Else：数据符合要求，进行下一步操作：

用约束条件对符合要求的数据进一步的筛选 ：满足约束条件则进行下一步要的操作，不满足约束条件的不进行下一步操作。

If （约束条件） - （属性条件） >0

则满足约束条件 进行下一步操作

Else 不满足

对于满足约束条件的属性所对应的物品放入背包中，此时背包中的物品的总数+1 ，背包中物品的总价值加上对应物品的价值。

Bag(i+1)

Maxvalues(0+price(i))

对所有的属性，用对应的约束条件进行判断，将满足约束条件的数据一提取出来，放入一个新的数组P中单独存储，这样就可以得到所有的满足约束条件的数据

在新的数组中，对筛选出来的数据进行进一步的筛选，选出满足条件的最大数据。

Maxvalues = max【P】

最后打印输出最大价值：

Print（maxvalues）

调试代码：

import xlrd

import numpy as np

import pandas as pd

from numpy import \* ###导入我们进行数据处理的时候需要的包

data = xlrd.open\_workbook("C:\\Users\\Administrator\\Desktop\\data.xlsx") #读取第一个表格中的数据

sheet = data.sheets()[0]

nrows = sheet.nrows #行数

m = sheet.nrows # 将表格的内容赋值给指定函数

for i in range(m):

print(sheet.row\_values(i)) ###采取一个for循环，将excel中的数据提取出来

data2= xlrd.open\_workbook("C:\\Users\\Administrator\\Desktop\\data2.xlsx") #读取第二个表格中的数据

sheet2 = data2.sheets()[0]

n = sheet2.nrows

print(sheet2.row\_values(0)) # 限定值，单独赋予check数组

price=sheet.row\_values(0) #价值 一共6个物品 每个物品对应10个属性

c0=sheet.row\_values(1)

c1=sheet.row\_values(2)

c2=sheet.row\_values(3)

c3=sheet.row\_values(4)

c4=sheet.row\_values(5)

c5=sheet.row\_values(6)

c6=sheet.row\_values(7)

c7=sheet.row\_values(8)

c8=sheet.row\_values(9)

c9=sheet.row\_values(10) #第N个限定条件的数值

check=sheet2.row\_values(0) #选取的以上的每行数值加起来小于本行对应的数值

print(check)

list\_data = [] #定义一个新的数组

#list\_data.append(price)

list\_data.append(c0)

list\_data.append(c1)

list\_data.append(c2)

list\_data.append(c3)

list\_data.append(c4)

list\_data.append(c5)

list\_data.append(c6)

list\_data.append(c7)

list\_data.append(c8)

list\_data.append(c9) #用append函数，将除了价格外的所有约束条件放在一个LIST里

y = pd.DataFrame(y)

check = pd.DataFrame(check) #转换一下数组的格式

x=6

v=0 #赋值两个数，X代表商品的总数，v代表价值，初始值为0

c11 =price[0] #定义一个数，代表最大价值

def bag(i): #定义一个bag函数

global v,c11,price,x #声明全局变量

if i > x-1:

if v>c11:

c11=v #判断，如果背包中放入的物品数量大于物品的总数或者背包中物品的价值大于物品的总价值的时候，FALSE

else:

p=0

p=(check[0]-y[i])>=0 #约束条件始终要大于0

if p.all():

check[0]=check[0]-y[i] #当放入的 物品满足约束条件的要求时候

v=v+price[i] #总价值加上对应的物品价值

bag(i+1) #背包中的物品数量+1

v=v-price[i]

check[0]=check[0]+y[i]

bag(i+1)

bag(0) #价值从高到低排序，第一个为最大价值

print(c11) #打印输出最大总价值