**中山大学计算机学院**

**人工智能**

**本科生实验报告**

**（2022学年春季学期）**

课程名称：Artificial Intelligence

|  |  |  |  |
| --- | --- | --- | --- |
| 教学班级 | **系统结构方向** | 专业（方向） | **计算机专业** |
| 学号 | **21307358** | 姓名 | **曾慧蕾** |

# 实验题目

实验1：罗马尼亚旅行问题

你是一名正在罗马尼亚旅行的游客，现在你急需从城市1赶到城市2搭乘航班，因此，你需要找到从城市1到城市2的最短路径。

罗马尼亚的地图信息文件：Romania.txt

文件第1行：城市数道路数文件；第2行到第24行，每行是：城市1名城市2名路程

请基于上周你编写的最短路径程序，扩展实现一个搜索罗马尼亚城市间最短路径的导航程序，要求：

•出发城市和到达城市由用户在查询时输入

•对于城市名称，用户可以输入全称，也可以只输入首字母，且均不区分大小写

•向用户输出最短路径时，要输出途经的城市，以及路径总路程

•输出内容在直接反馈给用户的同时，还需追加写入一个文本文件中，作为记录日志

•为提升代码灵活性，你应在代码中合理引入函数和类（各定义至少一个）

•此外，将你定义的一些函数和类，存储在独立的模块文件中

分析罗马尼亚旅行问题的实验结果，至少展示以下城市间的路径：

•Arad → Bucharest

•Fagaras→ Dobreta

•Mehadia→ Sibiu

# 实验内容

1. 算法原理

使用dijkstra算法，记录从同一个顶点出发到其余各个顶点的最小距离

1. 伪代码

读入文件信息

#建图

For I in range(n):

For j in range(n):

建立一个n×n大小的记录顶点之间关系的矩阵arr，两点之间若有关系则其值为边的权值 若无关系则值为99

以起点begin为目标行 复制arr[begin]

建立访问数组visit 除起点外其余点都为0 表示未访问

For I in range(n):

For j in range(n):

找到low中最小值，记录最小值及其下标

根据记录的下标值令其访问数组置1

For j in range(n):

比较从新到达的顶点到其余各点的距离是否小于原顶点到达其余各点的距离，若是，则：

更新每一个点的距离

更新途经的地点

所有最小值已记录完毕 输出low[end]

写入文件

1. 关键代码展示（带注释）

content=read(file)

arr=[]

low=[]

num\_to\_city={} #用于将数字转换为城市名

letter={} #用于将城市名转换为数字

module.create\_map(content,arr,low,num\_to\_city,letter) #建图

list=input().split()

begin=letter[list[0][0].lower()]

end=letter[list[1][0].lower()] #转小写

for i in range(0,len(low)):

    low[i].distant=arr[begin][i]

low[i].way=num\_to\_city[begin]#初始化low列表 同时包括距离和途径地

list=content[0].split()

m=int(list[0]) #节点数

visit=[value\*0 for value in range(0,m)]

visit[begin]=1 #初始化visit列表

min=1000

k=0

for i in range(0,m):

    min=1000

    for j in range(0,m):

        if visit[j]==0 and low[j].distant<min:

            min=low[j].distant

            k=j

    visit[k]=1

    for j in range(0,m):

        if low[k].distant+arr[k][j]<low[j].distant:

            low[j].distant=low[k].distant+arr[k][j] #更新路途

            low[j].way=low[k].way+" → "+num\_to\_city[k] #更新途径地

    if k==end:

        break

low[end].way=low[end].way+" → "+num\_to\_city[end] #在最后补上终点名

print(low[end].way.title(),low[end].distant)

1. 创新点&优化（如果有）

直接用letter和num\_to\_city实现城市名与数组下标的转换，无需通过反复读文件来实现二者的转换，节省时间

# 实验结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）







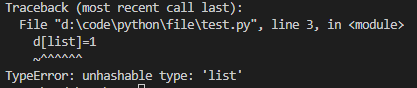
实验结果如上所示，包含了•Arad → Bucharest、•Fagaras→ Dobreta、•Mehadia→ Sibiu的路径

2. 评测指标展示及分析（机器学习实验必须有此项，其它可分析运行时间等）

**|-----------如有优化，请重复1，2，分析优化后的算法结果-----------------------|**

# 思考题

**1. 当我尝试以一个列表作为字典的key时，发生了如下现象：**

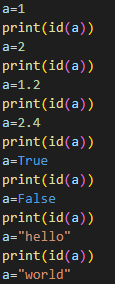
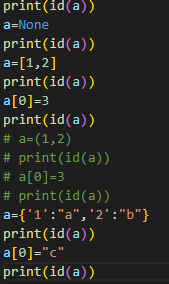
报错为：unhashable

但当我用一个元组作为字典的key时：

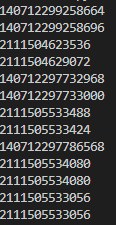
 

未报错

**2.**

 涉及到元组的内容修改无法执行，故元组为不可变数据类型

以上代码运行结果如下

可见除了列表和字典外，其余数据类型对应的内存地址都发生了变化

故：不可变数据类型有：整型 浮点型，布尔型，字符串，空串，元组

可变数据类型有：列表，字典

# 参考资料

**超算习堂-人工智能理论与实践-课时3：第二讲 实验课**

PS：可以自己设计报告模板，但是内容必须包括上述的几个部分，不需要写实验感想

六：课堂练习题

9.魔术师，了不起的魔术师与不变的魔术师：

a)创建一个包含魔术师名字的列表，并将其传递给show\_magicians()函数，这个函数打印列表中每个魔术师的名字。

b)编写一个名为make\_great()的函数，对魔术师列表进行修改，在每个魔术师的名字中都加入字样“the Great”。调用show\_magicians()，确认魔术师列表确实变了。

c)在调用make\_great()时，向它传递魔术师列表的副本。由于不想修改原始列表，请返回修改后的列表，并将其存储到另一个变量中。分别对这两列表调用show\_magicians()，确认一个列表包含的是原来的魔术师名字，而另一个列表包含的是添加了字样“the Great”的魔术师名字。

#homework9

magicians=['doudz','ahrenl','cantian']

def show\_magicians(magicians):

    print("Magicians:")

    for i in range(len(magicians)):

        print("-",magicians[i])

def make\_great(magicians):

    for i in range(len(magicians)):

        magicians[i]="the Great "+magicians[i]

    show\_magicians(magicians)

show\_magicians(magicians[:])

make\_great(magicians)

10.餐馆、就餐人数与冰淇淋小店：

a)创建一个名为Restaurant的类，其方法\_\_init\_\_()设置两个属性：restaurant\_name和cuisine\_type。创建一个名为describe\_restaurant()的方法和一个名为open\_restaurant()的方法，其中前者打印前述两项信息，而后者打印一条消息，指出餐馆正在营业。

b)添加一个名为number\_served的属性，并将其默认值设置为0。根据这个类创建一个名为restaurant的实例；打印有多少人在这家餐馆就餐过，然后修改这个值并再次打印它。添加一个名为set\_number\_served()的方法，它让你能够设置就餐人数；调用这个方法并向它传递一个值，然后再次打印这个值。添加一个名为increment\_number\_served()的方法，它让你能够将就餐人数递增；调用这个方法并向它传递一个这样的值：你认为这家餐馆每天可能接待的就餐人数。

c)冰淇淋小店是一种特殊的餐馆。编写一个名为IceCreamStand的类，让它继承Restaurant类。添加一个名为flavors的属性，用于存储一个由各种口味的冰淇淋组成的列表。编写一个显示这些冰淇淋的方法。创建一个IceCreamStand实例，并调用这个方法。

#homework10

class Restaurant():

    def \_\_init\_\_(self,restaurant\_name,cuisine\_type):

        self.restaurant\_name=restaurant\_name

        self.cuisine\_type=cuisine\_type

        self.number\_served=0

    def describe\_restaurant(self):

        print("Restaurant's information:")

        print("Name:",self.restaurant\_name)

        print("Cuisine type:",self.cuisine\_type)

    def open\_restaurant():

        print("The restaurant is opening")

    def set\_number\_served(self,num):

        self.number\_served=num

    def increment\_number\_served(self,num):

        self.number\_served+=num

my\_restaurant=Restaurant('萨莉亚','意大利面')

print(my\_restaurant.number\_served)

my\_restaurant.number\_served=114514

print(my\_restaurant.number\_served)

my\_restaurant.set\_number\_served(1919)

print(my\_restaurant.number\_served)

my\_restaurant.increment\_number\_served(810)

print(my\_restaurant.number\_served)

class IceCreamStand(Restaurant):

    def \_\_init\_\_(self, restaurant\_name, cuisine\_type,flavors):

        super().\_\_init\_\_(restaurant\_name, cuisine\_type)

        self.flavors=flavors

    def describe\_restaurant(self):

        print("IceCreamStand:")

        super().describe\_restaurant()

        print("flavors:",self.flavors)

my\_icecreamstand=IceCreamStand('麦当劳','麦旋风','奥利奥风味')

my\_icecreamstand.describe\_restaurant()

11.访问Project Gutenberg（https://gutenberg.org/），并找一些你想分析的图书。下载这些作品的文本文件或将浏览器中的原始文本复制到文本文件中。编写一个程序读入这些文本文件，分别对每个文本文件进行初步的统计分析：a)统计文本中包含多少个单词（提示：使用字符串方法split()与函数len()）b)统计单词“the”出现了多少次（提示：使用字符串方法count()与lower()）

homework11

def analysis(filename):

    try:

        with open(filename,'r') as file\_object:

            contents=file\_object.read()

    except FileNotFoundError:

        print("Sorry,the file",filename," does not exist.")

    else:

        length=len(contents.split())

        print("There are",length,"words in total")

        num\_the=contents.lower().count('the')

        print("There are",num\_the,"'the' words in total")

file='D:\\code\\python\\homework\_needs\\bible.txt'

analysis(file)

12.请你建立一个字典，并以一个字典中不存在的键访问该字典。a)观察报错信息，识别程序发生了哪种异常。b)改写代码，使程序可以处理这种异常。

#homework12

game={"hollow\_knight":"Team Cherry",

      "the\_binding\_of\_issac":"Edmund McMillen,Florian Himsl",

      "elden\_ring":"Miyazaki Hidetaka"}

try:

    search=game["silksong"]

except KeyError:

    print("this game does not exist!")

else:

    print("The producer of this game:",search)