中山大学计算机学院

人工智能

本科生实验报告

课程名称：Artificial Intelligence

学号	22320131	姓名	韦百强

一、 实验题目

1. 二分查找

   给定一个 n 个元素有序的(升序)整型数组 nums 和一个目标值 target ，写一个函数 BinarySearch 搜索 nums 中的 target ，如果目标值存在返回下标，否则返回 -1。

2. 矩阵加法,乘法

   给定两个 n × n 的整型矩阵 A 和 B,写两个函数 MatrixAdd 和 MatrixMul,分别得出这两个矩阵加法和乘法的结果

   两个矩阵的数据类型为嵌套列表,即 list[list],且满足 len(list)==n，len(list[01)==n.

   注意不要打乱原矩阵 A 和 B 中的数据。

3. 字典遍历

   给定非空字典 dict1,其键为姓名,值是学号.写一个函数 ReverseKeyValue 返回另个字典,其键是学号,值是姓名.

   例如, dict1={'Alice':'001','Bob':'002},则 ReverseKeyValue(dict1)返回的结果是{'001':'Alice'，'002':'Bob'}

4. Student_data 类

   给定 student data. txt 文本文件，每一行是一名学生的信息，从左到右分别是该学生的姓名，学号，性别和年龄，封装一个 StuData 类，实现 __init__函数打开文件并提取数据，SortData 函数按照特定数据排序，ExportFile 函数生成新的 txt 文件并填入数据，AddData 函数添加新的学生信息。

二、 实验内容

5. 算法原理

第一题：

对于升序的序列，采用经典二分查找算法：

1. 初始化比较区间，left = 0， right = n – 1；
2. 寻找中间值 mid = (left + right) // 2
3. 对中间索引 mid 的值和 target 进行比较：
   a) 若 mid 索引位置的值小于 target， left = mid + 1
   b) 若 mid 索引位置的值大于 target， right = mid – 1
   c) 若相等，则返回 mid
4. 若第三步没达到 c)，则重复第三步，直到 left > right
5. 若最后 left > right，则返回-1，表示找不到 target

第二题：

设结果矩阵设置为 add 和 mul：

1. 矩阵加法，add[i][j] = A[i][j] + B[i][j]
2. 矩阵乘法，k 从 0 到 n – 1：mul[i][j] += A[i][k] + B[k][j]，即 A 的 i 行与 B 的 j 列的各项相乘再相加

第三题：

对给定的非空字典用 items 函数获取键和值，循环对组键-值进行反转。

第四题：

封装一个类.

1. 循环读入文件中的每行信息，去除每行信息的末尾换行符，以空格为分隔符进行字符串分割，赋值给一个列表，在总的学生信息 self.data 中添加这条信息。
2. AddData，直接以列表形式添加信息。
3. SortData，输入信息关键字，以关键字所在的信息为排序依据排序，然后更新总信息。
4. ExportFile，输入新的文件名，循环写入一个新的文件。

6. 关键代码展示（可选）

第一题：

```
    while (left <= right):
        mid = (left + right) // 2
        if target < nums[mid]:
            right = mid - 1
        elif target > nums[mid]:
            left = mid + 1
        else:
            return mid
```

第二题：

表示矩阵的相乘关键代码：

```
    for i in range(n):
        for j in range(n):
            for k in range(n):
                result[i][j] += A[i][k] * B[k][j]
```

第三题：

```
def ReverseKeyValue(dict1):
    # 我写的函数，用来反转给定非空字典的键值
    dict2 = {}

    for key, value in dict1.items():
        dict2[value] = key

    return dict2
```

第四题：

```
import os
class StuData:
    def __init__(self, file_name):
        self.data = []

        with open(file_name, 'r') as student:
            for line in student:
                one_of_student = line.strip().split()
                age = int(one_of_student[3])
                student_new = [one_of_student[0], one_of_student[1],
one_of_student[2], age]
                self.data.append(student_new)

    def AddData(self, name, stu_num, gender, age):
        new_stu_data = [name, stu_num, gender, age]
        self.data.append(new_stu_data)

    def SortData(self, form__):
        if form__ == 'name':
            self.data.sort(key=lambda x: x[0])
        elif form__ == 'stu_num':
            self.data.sort(key=lambda x: x[1])
        elif form__ == 'gender':
            self.data.sort(key=lambda x: x[2])
        elif form__ == 'age':
```

```
            self.data.sort(key=lambda x: int(x[3]))

    def ExportFile(self, out_file):
        with open(out_file, 'w') as outfile:
            for student in self.data:
                outfile.write(f"{student[0]} {student[1]} {student[2]}\
{student[3]}\n")
```

7. 创新点&优化（如果有）

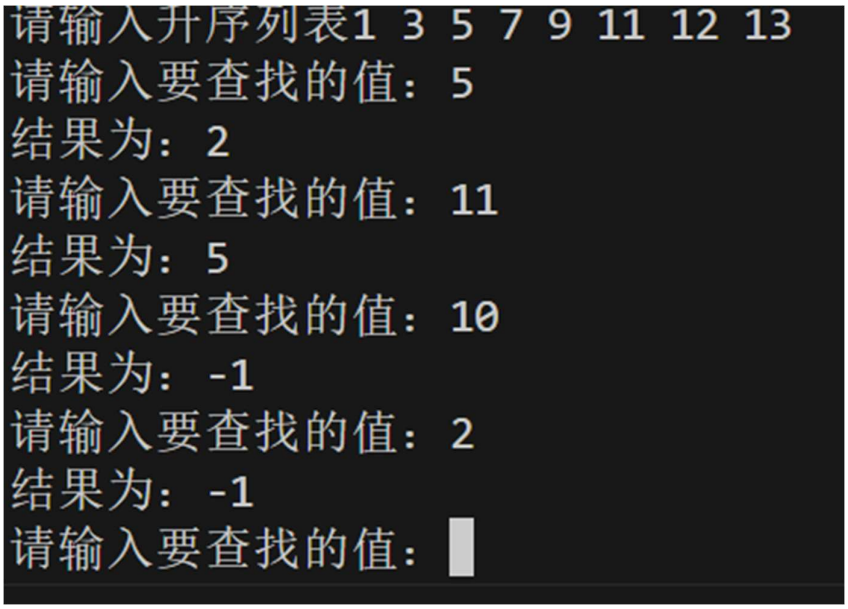
第三题：

对于我写的这个简单函数，只能对 one-to-one 的字典有效，如果多个键映射的值相同，那么反转之后，只会保留字典中排最后的键-值。我查阅资料，发现用 collections 类的 defaultdict 函数可以解决这个问题。但是直接输出结果（反转后的字典），会带有 defaultdict(<class 'list'>)，于是我又查阅资料，发现嵌套字典可以解决问题。

三、 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）

第一题：



第二题：

```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = [[11, 12, 13], [14, 15, 16], [17, 18, 19]]
相加结果为: [[12, 14, 16], [18, 20, 22], [24, 26, 28]]
相乘结果为: [[90, 96, 102], [216, 231, 246], [342, 366, 390]]
```

第三题：

```
test1:  {'男': '1', '女': 'g'}
test2:  {'小明': 'name', 18: 'age', '男': 'gender', '001': 'id'}
test3:  defaultdict(<class 'list'>, {'男': ['name', 'age', 'id', '1'], '女': ['g']})
test4:  {'男': ['name', 'age', 'id', '1'], '女': ['g']}
```

第四题：

加入新学生 weibaiqiang 数据后按 age 排序：

new_stu_data.txt

```
1   Leo 092 M 17
2   Aaron 243 M 18
3   Sam 230 F 18
4   Eric 249 M 19
5   Alex 812 M 19
6   Ruth 942 M 19
7   Cynthia 920 F 19
8   weibaiqiang 22320131 M 19
9   Beryl 091 F 20
10
```

加入 weibaiqiang 后按 gender 排序：

new_stu_data.txt

```
1   Sam 230 F 18
2   Beryl 091 F 20
3   Cynthia 920 F 19
4   Aaron 243 M 18
5   Eric 249 M 19
6   Alex 812 M 19
7   Leo 092 M 17
8   Ruth 942 M 19
9   weibaiqiang 22320131 M 19
10
```

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）

第一题：

简单经典的二分查找，时间复杂度是 O(logn)，最优是 O(1)，最坏是 O(n) [上网检查，最坏是 O(logn)才对]

第二题:

时间复杂度是 O(n^3)，因为乘法三层循环。

第三题:

时间复杂度应该是 O(n)，其中 n 为字典内的映射组数

第四题:

时间复杂度平均应该是 O(nlogn)，主要体现在 sort 函数的使用，因为这是一个 Tim 排序（网上搜的），其他部分都是线性时间 O(n)或者 O(1)

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

四、 参考资料

将参考的代码,算法思路的来源表明在此处(可以是网址，参考文献等)

第三题我参考了资料：

1. 关于相同值不同键的情况，参考 https://blog.csdn.net/weixin_46707326/article/details/117387329
2. 关于如何不 print 出类型，参考 https://www.thinbug.com/q/48823942

第四题我参考了资料：

1. 关于去除字符串首尾（主要是尾部）的空格和换行符，并分割字符串赋值给列表，参考 https://blog.csdn.net/Woo_home/article/details/88588197
2. 关于 Python 对于文件的基本操作，参考 https://zhuanlan.zhihu.com/p/675090788
3. 关于 sort 函数的使用，尤其是对嵌套列表特定元素排序的方法，参考 https://blog.csdn.net/iprobobo/article/details/122713695