人工智能实验报告

姓名：孟衍璋 学号：16337183

一、 请描述一下本次实验的主要内容:

实现 A*算法，定义自己的启发式函数，计算无向图中的最短路径问题。实验给定了一张地图，需要做的是将这张地图的信息导入，然后用当前已经走过的距离加上当前点到终点的直线距离定义启发式函数。相当于将 BFS 与贪心算法结合在了一起。

二、 Python 实现代码

```
import numpy

a,b,c,d,e,f,g,h,i,l,m,n,o,p,r,s,t,u,v,z = range(20)

city_map = [
    {s:140, t:118, z:75}, # a
    {f:211, g:90, p:101, u:85}, # b
    {d:120, p:138, r:146}, # c
    {c:120, m:75}, # d
    {h:86}, # e
    {b:211, s:99}, # f
```

```
    {b:90}, # g
    {e:86, u:98}, # h
    {n:87, v:92}, # i
    {m:70, t:111}, # l
    {d:75, l:70}, # m
    {i:87}, # n
    {s:151, z:71}, # o
    {b:101, c:138, r:97}, # p
    {c:146, p:97, s:80}, # r
    {a:140, f:99, o:151, r:80}, # s
    {a:118, l:111}, # t
    {b:85, h:98, v:142}, # u
    {i:92, u:142}, # v
    {a:75, o:71}, # z
]

straight_line_distance_to_b = {a:366, b:0, c:160, d:242, e:161, f:178, g:77, h:151, i:226, l:244, m:241, n:234, o:380, p:98, r:193, s:253, t:329, u:80, v:199, z:374}

num_to_point = {0:'a', 1:'b', 2:'c', 3:'d', 4:'e', 5:'f', 6:'g', 7:'h', 8:'i', 9:'l', 10:'m', 11:'n', 12:'o', 13:'p', 14:'r', 15:'s', 16:'t', 17:'u',
```

```
18:'v', 19:'z'}

def heuristic(cost_so_far, current_point):
    return cost_so_far +
straight_line_distance_to_b[current_point]

start_point = a
end_point = b
cost_so_far = 0
print("start form:", num_to_point[start_point])
current_point = start_point
route = list()

while(current_point != end_point):
    last_point = current_point
    temp_point = current_point
    h = float("inf")
    for i in city_map[current_point]:
        if(i in route):
            pass
        else:
            search = heuristic((cost_so_far +
```

```
city_map[current_point][i]), i)
            if(search < h):
                h = search
                temp_point = i
    route.append(temp_point)
    current_point = temp_point
    cost_so_far += city_map[last_point][current_point]

for i in route:
    if(i != end_point):
        print("next:", num_to_point[i])
    else:
        print("end:", num_to_point[end_point])
```

三、 介绍一下你所实现代码采用什么样的数据结构？分析一下算法的计算复杂度。

本次实验用 python 完成，存储地图用的是加权邻接字典。最外层是一个 list，里面的元素是一个个 dict，其中每个 dict 存储了相应城市的邻居还有它们之间的距离。之后再用一个 dict 存储了每个城市距离城市 b 的距离（因此也只能够计算其他城市到 b 的最短路径）。BFS 很适合找到最短的路径，但它浪费时间探索不方便的方向。贪心算法的最佳第一搜索探索有

前途的方向，但它可能找不到最短的路径。A*算法在保证速度的同时，也能找到最优路径。算法的计算复杂度优于 BFS。