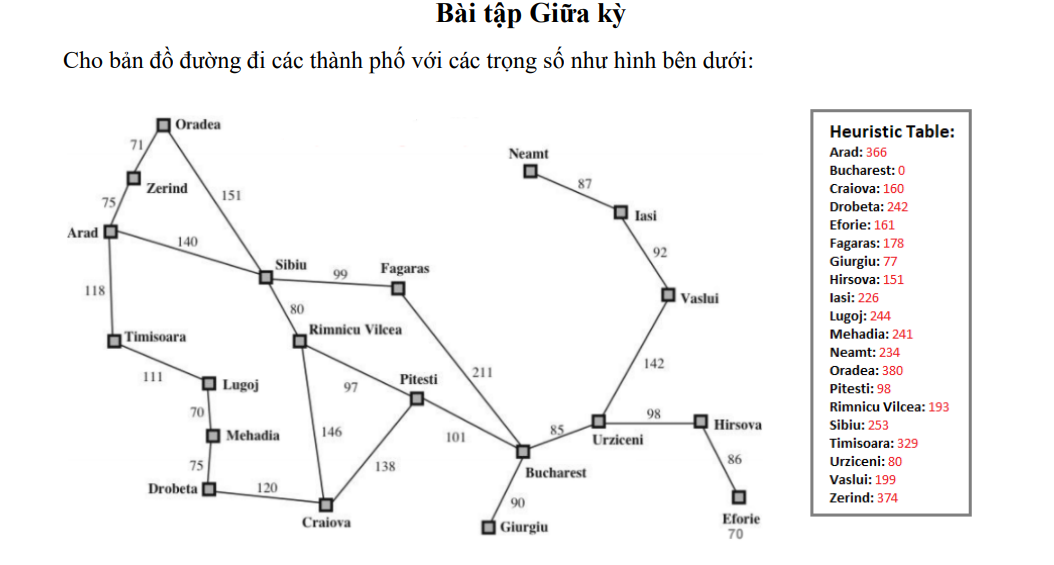
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên | Mssv | Tên | Mssv | Môn học |
| Đổ Thị Hồng Nhung | 1824801030045 | Nguyễn Hoài Nam | 1824801030027 | Trí tuệ nhân tạo |



Hàm đánh giá heuristic cho trong bảng là khoảng cách theo đường thẳng từ các thành phố tới thành phố Bucharest. Hãy viết chương trình tìm đường đi đến thành phố Bucharest theo yêu cầu sau:

Đề 4: Thành phố bắt đầu: Zerind Phương pháp tìm kiếm theo chiều sâu (Depth-first search)

1. **Mô tả bài toán**

– Đây là bài toán tìm đường đi bằng cách duyệt theo chiều sâu.

– Xuất phát từ 1 đỉnh và đi mãi cho đến khi không thể đi tiếp, sau đó đi về lại đỉnh đầu.  
Trong quá trình quay lại:

+ nếu gặp đường đi khác thì đi cho đến khi không đi tiếp được nữa

+ nếu không tìm ra đường đi nào khác thì ngừng việc tìm kiếm.

– Trong quá trình đi đến đỉnh khác, thuật toán sẽ lưu lại đỉnh cha vừa đi qua để khi đi ngược lại từ đỉnh Kết thúc đến đỉnh Xuất phát, ta có thể xem được đường đi từ đỉnh Kết thúc đến đỉnh Bắt Đầu.

– Sở dĩ thuật toán này tìm được đường đi là nhờ vào cơ chế tô màu và lưu đỉnh cha. Quá trình tô màu khiến 1 đỉnh không thể xét 2 lần trở lên và có thể xem được đường đi từ đỉnh Kết Thúc đến đỉnh Xuất phát dựa vào việc lưu đỉnh cha.

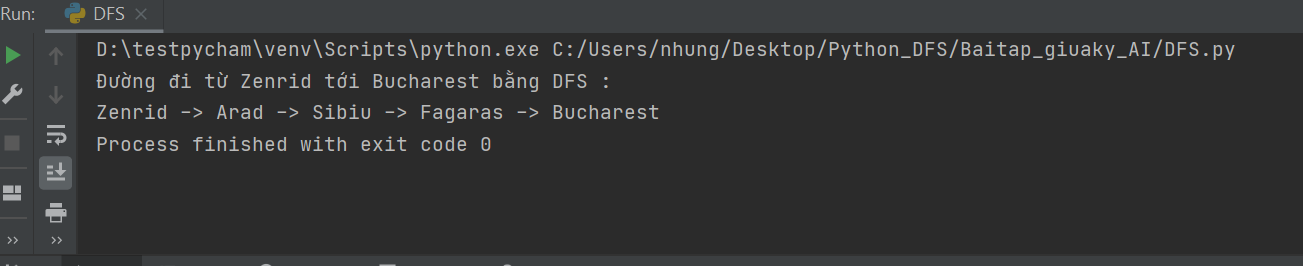
1. **Phương pháp sử dụng.**

* Gồm các bước:

1. Cho đỉnh xuất phát vào open.
2. Nếu open rỗng thì tìm kiếm thất bại, kết thúc việc tìm kiếm.
3. Lấy đỉnh đầu trong open ra và gọi đó là O. Cho O vào closed.
4. Nếu O là đỉnh đích thì tìm kiếm thành công, kết thúc việc tìm kiếm.
5. Tìm tất cả các đỉnh con của O không thuộc open và closed cho vào đầu open (theo ưu tiên trọng số nhỏ hơn nếu có).
6. Trở lại bước 2.
7. **Cài đặt hàm**
8. Hàm giải thuật tìm kiếm theo chiều sâu:

def dfs(graph, node, visited):  
 if node not in visited:  
 visited.append(node)  
 for n in graph[node]:  
 dfs(graph,n, visited)  
 return visited  
  
visited = dfs(graph1,'Oraden', [])  
count = 0  
for n in visited:  
 if n == 'Zenrid' :  
 diembatdau = count + 1  
 if n == 'Bucharest' :  
 diemketthuc = count  
 count+=1  
print('Đường đi từ Zenrid tới Bucharest bằng DFS :')  
print(visited[diembatdau-1],end='')  
while diembatdau <= diemketthuc :  
 print(' -> '+visited[diembatdau],end='')  
 diembatdau+=1

1. **Kết quả**
2. Kết quả chạy trên python.



1. Kết quả giải tay :

|  |  |  |  |
| --- | --- | --- | --- |
| Lần lặp | Đỉnh | Open | Close |
| 0 |  | Z[374] | {} |
| 1 | Z[374] | {a[366],o[380]} | z |
| 2 | A[366] | {s[253],t[329],o[380]} | a |
| 3 | S[253] | {f[178],r[193],o[380]} | s |
| 4 | F178] | {b[0], r[193],o[380]} | f |
| 5 | B[0] | Close; |  |

=> đường đi từ Zenrid => Bucharest là : Z => A =>S =>F =>B;

Tổng độ dài đường đi là : ZA+AS+SF+FB=75+140+99+211=525