

# NHẬP MÔN TRÍ TUỆ NHÂN TẠO

Tên: Trần Kim Khanh

Mssv:21110318

Bài tập thực hành 3

1 Bài toán tìm đường đi ngắn nhất :

+ Thuật toán Greedy-Best-First Search (GBFS):

- Mục tiêu: Dựa vào tri thức bài toán (heuristic) để tìm đường đi một cách tham lam.
- Nguyên tắc hoạt động:
  - Sử dụng hàm heuristic  $h(n)$ , ước tính chi phí từ trạng thái hiện tại đến mục tiêu.
  - Mỗi lần mở rộng trạng thái, chọn trạng thái có giá trị  $h(n)$  nhỏ nhất.
- Đặc điểm:
  - Chỉ xét chi phí ước lượng  $h(n)$ , không quan tâm đến chi phí thực tế đã đi qua.
  - Không đảm bảo tìm được đường đi ngắn nhất.
  - Phù hợp khi heuristic cung cấp thông tin chính xác về bài toán.

Mã giả:

```

Hàm GBFS(startNode, heuristics, graph, goalNode):
    priorityQueue ← Tạo hàng đợi ưu tiên rỗng
    Thêm (heuristics[startNode], startNode) vào priorityQueue

    path ← Danh sách rỗng

    Trong khi priorityQueue không rỗng:
        current ← Lấy phần tử có ưu tiên cao nhất từ priorityQueue
        Thêm current vào path

        Nếu current là goalNode:
            Dừng vòng lặp

        priorityQueue ← Tạo hàng đợi ưu tiên mới (xóa các nút cũ)

        Với mỗi nút i trong danh sách lân cận của current (graph[current]):
            Nếu i[0] không có trong path:
                Thêm (heuristics[i[0]], i[0]) vào priorityQueue

    Trả về path

```

Chạy tay thuật toán GBFS:

Band đầu:

đầu tiên với  $h(n)$  được xác định như sau:

$h(\text{Arad}) = 366$	$h(\text{Hirsova}) = 0$	$h(\text{Rimnicu Vilcea}) = 193$
$h(\text{Bucharest}) = 20$	$h(\text{Iasi}) = 226$	$h(\text{Sibiu}) = 253$
$h(\text{Craiova}) = 160$	$h(\text{Lugoj}) = 244$	$h(\text{Timisoara}) = 329$
$h(\text{Drobeta}) = 242$	$h(\text{Mehadia}) = 241$	$h(\text{Urziceni}) = 10$
$h(\text{Eforie}) = 161$	$h(\text{Neamt}) = 234$	$h(\text{Vaslui}) = 199$
$h(\text{Fagaras}) = 176$	$h(\text{Oradea}) = 380$	$h(\text{Zerind}) = 374$
$h(\text{Giurgiu}) = 77$	$h(\text{Pitesti}) = 100$	

Đi từ Arad(1) đến Hirsova(8):

Bắt đầu:

đi từ **Arad** , queue=[(**sibiu : 253**) , (Timisoara: 329) , (Zerind:374)], path[Arad]

đi từ **sibiu** , queue=[( **Fagaras : 176**), (Rimnicu Vilcea:193), (Timisoara: 329) , (Zerind:374)], path[Arad-> sibiu]

đi từ **Fagaras**, queue=[( **Bucharest: 20**), (Rimnicu Vilcea:193), (Timisoara: 329)] , (Zerind:374)], path[Arad-> sibiu-> Fagaras]

đi từ **Bucharest** , queue=[( **Urziceni: 10**), (Giurgiu:77),(Rimnicu Vilcea:193) , (Timisoara: 329)] , (Zerind:374)], path[Arad-> sibiu-> Fagaras-> Bucharest]

đi từ **Urziceni**, queue=[( **Hirsova: 10**), (Giurgiu:77),(Rimnicu Vilcea:193),( Vaslui :199) , (Timisoara: 329)] , (Zerind:374)], path[Arad-> sibiu-> Fagaras-> Bucharest-> Urziceni]

**Hirsova đã tới đích** path[Arad-> sibiu-> Fagaras-> Bucharest-> Urziceni-> Hirsova ]

Chạy code :

```
Nhap dinh bat dau: 1
Nhap dinh ket thuc: 8
GBFS => ['Arad', 'Sibiu', 'Fagaras', 'Bucharest', 'Urziceni', 'Hirsova']
```

Kết quả đúng như mong đợi

## 2 Thuật toán A\*:

Thuật toán A\* (A-star) là một thuật toán tìm kiếm được sử dụng rộng rãi để tìm đường đi ngắn nhất trong không gian trạng thái, kết hợp giữa phương pháp tìm kiếm theo chiều rộng và tìm kiếm theo chiều sâu. Thuật toán này sử dụng một hàm đánh giá để quyết định thứ tự các trạng thái cần khám phá.

### Cách thức hoạt động của thuật toán A\*:

#### 1. Hàm đánh giá:

- Thuật toán A\* sử dụng một hàm đánh giá  $f(n)$  để đánh giá mỗi nút  $n$ .
- Hàm này bao gồm hai phần:
  - **$g(n)$**  Chi phí từ nút gốc đến nút  $n$ .
  - **$h(n)$**  Heuristic, tức là chi phí ước lượng từ nút  $n$  đến nút đích.
- Hàm đánh giá là:  $f(n) = g(n) + h(n)$  Trong đó:
  - **$g(n)$** : Chi phí thực tế từ nút gốc đến nút  $n$ .
  - **$h(n)$** : Heuristic ước tính chi phí từ nút  $n$  đến nút đích.

#### 2. Điều kiện Heuristic:

- Hàm heuristic  $h(n)$  phải là "chấp nhận được" (admissible), tức là không bao giờ ước lượng cao hơn chi phí thực tế từ nút  $n$  đến đích. Điều này đảm bảo thuật toán A\* luôn tìm ra được đường đi ngắn nhất.

Mã giả:

## Procedure Astar-Search

### Begin

1. Đặt OPEN chỉ chứa  $T_0$ . Đặt  $g(T_0) = 0$ ,  $h(T_0) = 0$  và  $f(T_0) = 0$ . Đặt CLOSE là tập rỗng.
2. Lặp lại các bước cho đến khi gặp điều kiện dừng
  - 2.a. Nếu OPEN rỗng: bài toán vô nghiệm, thoát.
  - 2.b. Ngược lại, chọn  $T_{\max}$  trong OPEN sao cho  $f(T_{\max})$  là nhỏ nhất
    - 2.b.1. Lấy  $T_{\max}$  ra khỏi OPEN và đưa  $T_{\max}$  vào CLOSE.
    - 2.b.2. Nếu  $T_{\max}$  là  $T_G$  (trạng thái đích) thì thoát và thông báo lời giải là  $T_{\max}$
    - 2.b.3. Nếu  $T_{\max}$  không phải là  $T_G$ . Tạo ra danh sách tất cả các trạng thái kế tiếp của  $T_{\max}$ .  
Gọi một trạng thái này  $T_k$ . Với mỗi  $T_k$ , làm các bước sau:
      - 2.b.3.1. Tính  $g(T_k) = g(T_{\max}) + \text{cost}(T_{\max}, T_k)$
      - 2.b.3.2. Nếu tồn tại  $T_{k'}$  trong OPEN trùng với  $T_k$ .  
Nếu  $g(T_k) < g(T_{k'})$  thì  
Đặt  $g(T_{k'}) = g(T_k)$   
Tính lại  $f(T_{k'})$   
Đặt  $\text{Cha}(T_{k'}) = T_{\max}$
      - 2.b.3.3. Nếu tồn tại  $T_{k'}$  trong CLOSE trùng với  $T_k$   
Nếu  $g(T_k) < g(T_{k'})$  thì  
Đặt  $g(T_{k'}) = g(T_k)$   
Tính lại  $f(T_{k'})$   
Đặt  $\text{Cha}(T_{k'}) = T_{\max}$
    - 2.b.3.4. Nếu  $T_k$  chưa xuất hiện trong cả OPEN lẫn CLOSE thì  
Thêm  $T_k$  vào OPEN  
Tính:  $f(T_k) = g(T_k) + h(T_k)$

Chạy tay

Open = {(Arad,g= 0,h = 0, f = 0)}

Close = {}

Do OPEN chỉ chứa có 1 thành phố nên thành phố này sẽ là thành phố tốt nhất. Nghĩa là ta chọn Tmax = Arad. Lấy Arad ra khỏi OPEN và đưa vào CLOSE.

Open = {}

Close = {(Arad,g= 0,h = 0, f = 0)}

Từ Arad có thể đi được đến 3 thành phố Sibiu, Timisoara và Zerind. Ta lần lượt tính

f, g và h của 3 thành phố này. Do cả 3 nút mới tạo ra này chưa có nút cha nên ban đầu

nút cha của chúng đều là Arad.

$$1) f(\text{Sibiu}) = g(\text{Arad}) + h(\text{sibiu}) = 140 + 253 = 393$$

cha(Sibiu) = Arad

$$2) f(\text{Timisoara}) = g(\text{Timisoara}) + h(\text{Timisoara}) = 118 + 329 = 447$$

cha(Timisoara) = Arad

$$3) f(\text{Zerind}) = g(\text{Zerind}) + h(\text{Zerind}) = 75 + 374 = 449$$

cha(Zerind) = Arad

Do Sibiu, Timisoara và Zerind đều không có trong cả OPEN và CLOSE nên ta thêm 3 nút này vào OPEN.

Open = {( Sibiu, g=140,h=253,f=393,cha=Arad), (Timisoara, g=118,h=329,f=447,cha=Arad), }, (Zerind, g=75,h=374,f=449,cha=Arad),  
}

Close = {(Arad,g= 0,h = 0, f = 0)}

Bước tiếp đi từ :

Tmax = Sibiu

Open = { (Timisoara, g=118,h=329,f=447,cha=Arad), (Zerind, g=75,h=374,f=449,cha=Arad)}

Close = {(Arad,g= 0,h = 0, f = 0),( Sibiu, g=140,h=253,f=393,cha=Arad),}

Từ **Sibiu** có thể đi được đến Arad, Fagaras, Oradea, R. Vilcea. Ta lần lượt tính h, g và f của các nút này.

Tính toán các bước như trên

Open = { (Timisoara, g=118,h=329,f=447,cha=Arad),  
(Zerind, g=75,h=374,f=449,cha=Arad),  
( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
( Oradea, g = 291, h = 380, f = 617, Cha = Sibiu),  
( R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu)}

Close = {(Arad,g= 0,h = 0, f = 0),( Sibiu, g=140,h=253,f=393,cha=Arad)}

Tmax = R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu

Từ R. Vilcea có thể đi được tới 3 thành phố là Craiova, Pitesti và Sibiu. Ta lần lượt tính các giá trị, g và f của 3 thành phố này.

Open = { (Timisoara, g=118,h=329,f=447,cha=Arad),  
(Zerind, g=75,h=374,f=449,cha=Arad),  
( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
( Oradea, g = 291, h = 380, f = 617, Cha = Sibiu),  
(Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
(Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea)  
}

Close = {(Arad,g= 0,h = 0, f = 0),

( Sibiu, g=140,h=253,f=393,cha=Arad),

(R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu)

}

Tmax =( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu)

Từ Fagaras ta có thể đi được tới Sibiu và Bucharest.

Open = { (Timisoara, g=118,h=329,f=447,cha=Arad),

(Zerind, g=75,h=374,f=449,cha=Arad),

( Oradea, g = 291, h = 380, f = 617, Cha = Sibiu),

(Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),

(Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),

(Bucharest, g = 450, h = 20, f = 470, Cha = Fagaras)

}

Close = {(Arad,g= 0,h = 0, f = 0),

( Sibiu, g=140,h=253,f=393,cha=Arad),

(R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),

( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu)

}

Từ tập OPEN, nút tốt nhất là Pitesti nên Tmax=Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea

Từ Pitesti ta có thể đi được đến R. Vilcea, Bucharest và Craiova.

Open = { (Timisoara, g=118,h=329,f=447,cha=Arad),

(Zerind, g=75,h=374,f=449,cha=Arad),



( Oradea, g = 291, h = 380, f = 617, Cha = Sibiu),  
 (Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
 (Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti)  
 }  
 Close = {(Arad,g= 0,h = 0, f = 0),  
 ( Sibiu, g=140,h=253,f=393,cha=Arad),  
 (R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
 ( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
 (Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea)  
 }

Trong tập OPEN, Bucharest có giá trị f nhỏ nhất nên Tmax = (Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti). Từ Bucharest ta có thể được tới 4 thành phố Pitesti, Fagaras, Giurgiu, và Urziceni.

Open = { (Timisoara, g=118,h=329,f=447,cha=Arad),  
 (Zerind, g=75,h=374,f=449,cha=Arad),  
 ( Oradea, g = 291, h = 380, f = 617, Cha = Sibiu),  
 (Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
 (Giurgiu, g =508, h = 77, f =585, Cha = Bucharest),  
 (Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest)  
 }  
 Close = {(Arad,g= 0,h = 0, f = 0),  
 ( Sibiu, g=140,h=253,f=393,cha=Arad),  
 (R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
 ( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),

(Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),  
(Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti)  
}

Trong tập OPEN, Timisoara có giá trị f nhỏ nhất nên Tmax =(Timisoara, g=118,h=329,f=447,cha=Arad). Từ Timisoara ta có thể được tới 2 thành phố Arad và Lugoj

Open = {  
(Zerind, g=75,h=374,f=449,cha=Arad),  
( Oradea, g = 291, h = 380, f = 617, Cha = Sibiu),  
(Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
(Giurgiu, g =508, h = 77, f =585, Cha = Bucharest),  
(Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest)  
(Lugoj, g= 229,h=244, f= 473 ,cha = Timisoara)  
}

Close = {(Arad,g= 0,h = 0, f = 0),  
( Sibiu, g=140,h=253,f=393,cha=Arad),  
(R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
(Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),  
(Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti),  
(Timisoara, g=118,h=329,f=447,cha=Arad)  
}

Trong tập OPEN, Zerind có giá trị f nhỏ nhất nên Tmax (Zerind, g=75,h=374,f=449,cha=Arad). Từ Zerind ta có thể được tới 2 thành phố Arad và Oradea . Tính toán và cập nhật Oradea lại trong Open

Open = {

(Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),

(Giurgiu, g =508, h = 77, f =585, Cha = Bucharest),

(Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest)

(Lugoj, g= 229,h=244, f= 473 ,cha = Timisoara),

(Oradea,g=146,h=380,f=526,cha = Zerind)

}

Close = {(Arad,g= 0,h = 0, f = 0),

( Sibiu, g=140,h=253,f=393,cha=Arad),

(R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),

( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),

(Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),

(Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti),

(Timisoara, g=118,h=329,f=447,cha=Arad),

(Zerind, g=75,h=374,f=449,cha=Arad)

}

Trong tập OPEN, Lugoj có giá trị f nhỏ nhất nên Tmax= (Lugoj, g= 229,h=244, f= 473 ,cha = Timisoara).Từ Lugoj ta có thể được tới 2 thành phố Timisoran và mehadia. Thêm mehadia vào open

Open = {

(Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
 (Giurgiu, g = 508, h = 77, f = 585, Cha = Bucharest),  
 (Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest),  
 (Oradea, g = 146, h = 380, f = 526, cha = Zerind),  
 (Mehadia, g = 299, h = 241, f = 540, cha = Lugoj),  
 }

Close = {(Arad, g = 0, h = 0, f = 0),  
 ( Sibiu, g = 140, h = 253, f = 393, cha = Arad),  
 (R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
 ( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
 (Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),  
 (Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti),  
 (Timisoara, g = 118, h = 329, f = 447, cha = Arad),  
 (Zerind, g = 75, h = 374, f = 449, cha = Arad),  
 (Lugoj, g = 229, h = 244, f = 473, cha = Timisoara)  
 }

Trong tập OPEN, Urziceni có giá trị f nhỏ nhất nên Tmax = (Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest). Từ Urziceni ta có thể được tới 3 thành phố vaslui và Hirsova, Bucharest. Thêm vaslui và Hirsova vào

Open = {  
 (Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
 (Giurgiu, g = 508, h = 77, f = 585, Cha = Bucharest),  
 (Oradea, g = 146, h = 380, f = 526, cha = Zerind),

(Mehadia,g=299,h=241,f=540,cha =Lugoj),  
 (Vaslui,g=645,h=199,f=844,cha = Urziceni),  
 (Hirsova,g=601,h=0,f = 601,cha = Urziceni),  
 }  
 Close = {(Arad,g= 0,h = 0, f = 0),  
 ( Sibiu, g=140,h=253,f=393,cha=Arad),  
 (R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
 ( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
 (Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),  
 (Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti),  
 (Timisoara, g=118,h=329,f=447,cha=Arad),  
 (Zerind, g=75,h=374,f=449,cha=Arad),  
 (Lugoj, g= 229,h=244, f= 473 ,cha = Timisoara),  
 (Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest)  
 }

Trong tập OPEN, Craiova có giá trị f nhỏ nhất nên Tmax= (Craiova, g = 366, h  
 = 160, f = 526, Cha = R.Vilcea).Từ Craiova ta có thể được tới 4 thành phố

Drobeta,Mehadia,pitestri, Riminicu Vilcea. Chỉ thêm được Drobeta vào  
 Open = {

(Giurgiu, g =508, h = 77, f =585, Cha = Bucharest),  
 (Oradea,g=146,h=380,f=526,cha = Zerind),  
 (Mehadia,g=299,h=241,f=540,cha =Lugoj),  
 (Vaslui,g=645,h=199,f=844,cha = Urziceni),

(Hirsova,g=601,h=0,f = 601,cha = Urziceni),  
 (Drobeta,g= 480,h=242,g=722,cha =Craiova),  
 (  
 }  
 Close = {(Arad,g= 0,h = 0, f = 0),  
 ( Sibiu, g=140,h=253,f=393,cha=Arad),  
 (R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
 ( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
 (Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),  
 (Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti),  
 (Timisoara, g=118,h=329,f=447,cha=Arad),  
 (Zerind, g=75,h=374,f=449,cha=Arad),  
 (Lugoj, g= 229,h=244, f= 473 ,cha = Timisoara),  
 (Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest),  
 (Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea)  
  
 }

Trong tập OPEN, Oradea có giá trị f nhỏ nhất nên  
 Tmax(Oradea,g=146,h=380,f=526,cha = Zerind),.Từ Oradea ta có thể được tới  
 2 thành phố Zerind và sibiu. Ta ko cập nhật gì hết

Open = {  
 (Giurgiu, g =508, h = 77, f =585, Cha = Bucharest),  
 (Mehadia,g=299,h=241,f=540,cha =Lugoj),

(Vaslui,g=645,h=199,f=844,cha = Urziceni),  
(Hirsova,g=601,h=0,f = 601,cha = Urziceni),  
(Drobeta,g= 480,h=242,g=722,cha =Craiova),

}

Close = {(Arad,g= 0,h = 0, f = 0),  
( Sibiu, g=140,h=253,f=393,cha=Arad),  
(R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
(Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),  
(Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti),  
(Timisoara, g=118,h=329,f=447,cha=Arad),  
(Zerind, g=75,h=374,f=449,cha=Arad),  
(Lugoj, g= 229,h=244, f= 473 ,cha = Timisoara),  
(Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest),  
(Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
(Oradea,g=146,h=380,f=526,cha = Zerind)

}

Trong tập OPEN, Mehadia có giá trị f nhỏ nhất nên  
Tmax=(Mehadia,g=299,h=241,f=540,cha =Lugoj),Từ Mehadia ta có thể được  
tới

2 thành phố Lugoj và Drobeta. Ta cập nhật Drobeta trong open

Open = {

(Giurgiu, g =508, h = 77, f =585, Cha = Bucharest),  
(Vaslui,g=645,h=199,f=844,cha = Urziceni),  
(Hirsova,g=601,h=0,f = 601,cha = Urziceni),  
(Drobeta,g= 374,h=242,g=612,cha =Mehadia),

}

Close = {(Arad,g= 0,h = 0, f = 0),  
( Sibiu, g=140,h=253,f=393,cha=Arad),  
(R.Vilcea, g = 220, h = 193, f = 413, Cha = Sibiu),  
( Fagaras, g = 239, h = 176, f = 415, Cha = Sibiu),  
(Pitesti, g = 317, h = 100, f = 417, Cha = R.Vilcea),  
(Bucharest, g = 418, h = 20, f = 438, Cha = Pitesti),  
(Timisoara, g=118,h=329,f=447,cha=Arad),  
(Zerind, g=75,h=374,f=449,cha=Arad),  
(Lugoj, g= 229,h=244, f= 473 ,cha = Timisoara),  
(Urziceni, g = 503, h = 10, f = 513, Cha = Bucharest),  
(Craiova, g = 366, h = 160, f = 526, Cha = R.Vilcea),  
(Oradea,g=146,h=380,f=526,cha = Zerind)  
(Mehadia,g=299,h=241,f=540,cha =Lugoj),

}



Trong tập OPEN, Giurgiu có giá trị f nhỏ nhất nên  $T_{max} = (\text{Giurgiu}, g = 508, h = 77, f = 585, \text{Cha} = \text{Bucharest})$ . Từ Giurgiu ta có thể được tới Bucharest. Không cập nhật gì hết

Open = {

(Vaslui,  $g=645, h=199, f=844, \text{cha} = \text{Urziceni}$ ),

(Hirsova,  $g=601, h=0, f = 601, \text{cha} = \text{Urziceni}$ ),

(Drobeta,  $g= 374, h=242, g=612, \text{cha} = \text{Mehadia}$ ),

}

Close = {(Arad,  $g= 0, h = 0, f = 0$ ),

( Sibiu,  $g=140, h=253, f=393, \text{cha}=\text{Arad}$ ),

(R.Vilcea,  $g = 220, h = 193, f = 413, \text{Cha} = \text{Sibiu}$ ),

( Fagaras,  $g = 239, h = 176, f = 415, \text{Cha} = \text{Sibiu}$ ),

(Pitesti,  $g = 317, h = 100, f = 417, \text{Cha} = \text{R.Vilcea}$ ),

(Bucharest,  $g = 418, h = 20, f = 438, \text{Cha} = \text{Pitesti}$ ),

(Timisoara,  $g=118, h=329, f=447, \text{cha}=\text{Arad}$ ),

(Zerind,  $g=75, h=374, f=449, \text{cha}=\text{Arad}$ ),

(Lugoj,  $g= 229, h=244, f= 473, \text{cha} = \text{Timisoara}$ ),

(Urziceni,  $g = 503, h = 10, f = 513, \text{Cha} = \text{Bucharest}$ ),

(Craiova,  $g = 366, h = 160, f = 526, \text{Cha} = \text{R.Vilcea}$ ),

(Oradea,  $g=146, h=380, f=526, \text{cha} = \text{Zerind}$ )

(Mehadia,  $g=299, h=241, f=540, \text{cha} = \text{Lugoj}$ ),

(Giurgiu,  $g = 508, h = 77, f = 585, \text{Cha} = \text{Bucharest}$ )

}

Trong tập OPEN, Giurgiu có giá trị f nhỏ nhất nên Tmax(Hirsova,g=601,h=0,f=601,cha = Urziceni ) ta thấy nó đã là đích cần tới nên kết thúc chương trình

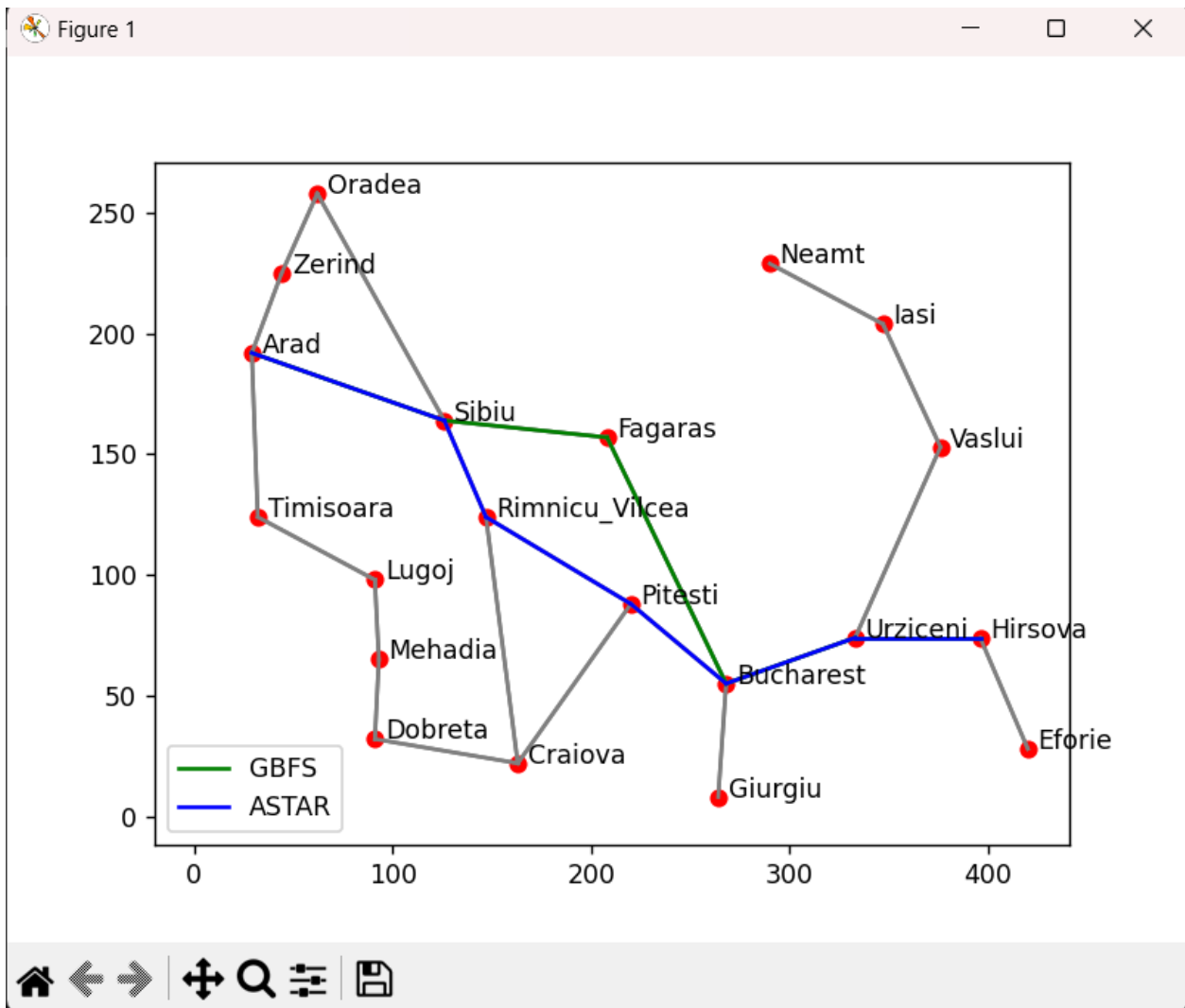
Đường đi sẽ là: Arad-> Sibiu-> Rimnicu\_Vilcea -> Pitesti ->Bucharest -> Urziceni-> Hirsova

Kết quả khi chạy code :

```
20 zerind
Nhap dinh bat dau: 1
Nhap dinh ket thuc: 8
GBFS => ['Arad', 'Sibiu', 'Fagaras', 'Bucharest', 'Urziceni', 'Hirsova']
ASTAR => ['Arad', 'Sibiu', 'Rimnicu_Vilcea', 'Pitesti', 'Bucharest', 'Urziceni', 'Hirsova']
```

Vậy ta thấy kết quả chạy tay và chạy code giống nhau.

Ảnh của kết quả :



\*Nhận xét :+ 2 thuật toán trên đúng với kết quả chạy tay của bản thân.

+ *Thuật toán A (A Star):\**

- Ưu điểm:
  - Tối ưu: A\* đảm bảo tìm được con đường ngắn nhất nếu heuristic là hợp lệ.
  - Cân bằng tốt: Kết hợp cả chi phí thực tế và dự đoán, giúp tìm kiếm hiệu quả.
- Nhược điểm:
  - Tốn bộ nhớ: Cần lưu trữ nhiều thông tin, có thể gây tốn bộ nhớ.

- Phụ thuộc vào heuristic: Nếu heuristic không chính xác,  $A^*$  sẽ chạy chậm hơn.

+Thuật toán Greedy Best-First Search (GBFS):

- Ưu điểm:
  - Nhanh: Tìm kiếm nhanh vì chỉ dựa vào heuristic, không cần xét đến chi phí đã đi qua.
  - Tiết kiệm bộ nhớ: Lưu trữ ít thông tin hơn so với  $A^*$ .
- Nhược điểm:
  - Không tối ưu: Có thể không tìm được đường đi ngắn nhất vì chỉ chú trọng vào heuristic.
  - Có thể mắc sai lầm: Vì chỉ xem xét gần nhất, nên đôi khi tìm được đường đi kém hơn.