

گزارش تمرین اول

برای این مساله هر state فقط دارای نام است که نام شهر های نقشه است.
action ها دارای id (برای شناسایی و تمایز از یکدیگر) و وزن (فاصله) اند.
Node ها دارای یک state و لینک لیست از از اکشنها است که از نود اول با این اکشنها به نود مربوطه رسیده است.
درکلاس problem :

initialState : حالت اولیه که همان Arad است را set میکند

actionset : در هر شهر فقط به شهرهای همسایه میتوان رفت

result : با گرفتن شهر فعلی و اکشن شهر مقصد را مشخص میکند

goalTest : state را با Vaslui مقایسه میکند

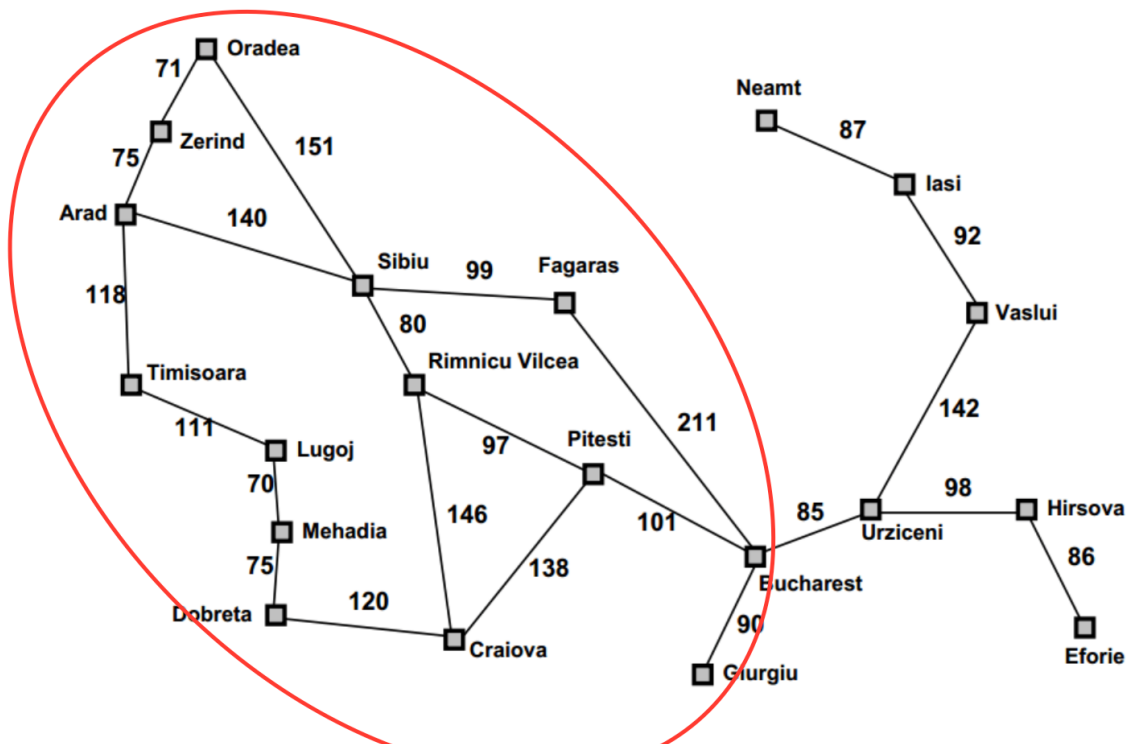
actionCost : مسافت یا cost هر اکشن را مشخص میکند.

pathCost : مسافت یا cost مجموعه ای از اکشنها را مشخص میکند.

Heuristic : برای نوشتن این تابع از تابع فاصله مستقیم بین Arad و Bucharest استفاده کردم.

تمام شهرهای این قسمت برای رفتن به Vaslui مجبورند از Bucharest عبور کنند. همچنین فاصله مستقیم بین Bucharest و Vaslui را هم داریم و چون cost هر کدام بیش از این عدد است پس جمع این دو عدد همواره از cost کوچکتر خواهد شد.

برای شهر های سمت راست عددی کمی کوچکتر از فاصله مستقیم در نظر گرفتم.



الگوریتم سطح اول درختی:

```
BFS tree :  
Visited Nodes: 195  
Expanded Nodes: 73  
Best Path: Arad Sibiu Fagaras Bucharest Urziceni Vaslui  
Path Cost: 677  
Total Time: 13006490794443  
Max Capacity for e and f: 120  
Max Capacity for path in nodes: 5
```

الگوریتم عمق اول محدود:

```
Limited DFS :  
Visited Nodes: 30  
Expanded Nodes: 11  
Best Path: Arad Sibiu Fagaras Bucharest Urziceni Vaslui  
Path Cost: 677  
Total Time: 12985493723702  
Max Capacity for e and f: 15  
Max Capacity for path in nodes: 5
```

الگوریتم A* :

```
AStar :  
Visited Nodes: 25  
Expanded Nodes: 8  
Best Path: Arad Sibiu Rimnicu Vilcea Pitesti Bucharest Urziceni Vaslui  
Path Cost: 645  
Total Time: 3537337  
Max Capacity for e and f: 13  
Max Capacity for path in nodes: 6
```

الگوریتم A* فقط مسیر بهینه را به ما داد.
الگوریتم سطح اول درختی نود های خیلی زیادی را بسط داده و الگوریتم A* کمترین نود را بسط داده است.
الگوریتم سطح اول درختی تعداد بازدید نودهایش هم به دلیل تولید نود تکراری زیاد است.