

**سوال اول (توضیحات کلی):**

در ابتدای اجرای برنامه، گراف از یک فایل به نام input خوانده می‌شود و به صورت لیست مجاورت ذخیره می‌شود. State های اولیه و هدف نیز همینجا مشخص می‌شوند. در این مرحله همچنین فایلی به نام heuristics خوانده می‌شود که هیوریستیک های داده شده در آن نوشته شده است. سپس در یک حلقه بی‌نهایت از کاربر ورودی گرفته می‌شود که همان دستورهای هستند که برنامه اجرا می‌کند. در اینجا کاربر مشخص می‌کند کدام جست و جو و با چه آرگمان‌هایی انجام شود و خروجی بلافاصله نمایش داده می‌شود.

**سوال اول (الگوریتم ها):**

- سطح اول درختی:

```
bfs tree
Arad,Timisoara,Lugoj,Mehadia,Dobreta,Craiova,Pitesti,Bucharest
path length: 7
path cost: 733
expanded: 7
visisted: 17
max nodes in memory: 11
```

- سطح اول گرافی:

```
bfs graph
Arad,Timisoara,Lugoj,Mehadia,Dobreta,Craiova,Pitesti,Bucharest
path length: 7
path cost: 733
expanded: 7
visisted: 17
max nodes in memory: 11
```

- عمق اول نامحدود درختی: این الگوریتم درختش بی‌نهایت می‌شود چون الگوریتم بین دو شهر در حلقه می‌افتد

```
dfs tree normal
Segmentation fault (core dumped)
amir@amir-UX310UQ:~/c++_workspace/AI_search_algorithms/systematic$
```

- عمق اول نامحدود گرافی: الگوریتم عمق اول را به صورت backtracking پیاده سازی کردم

```
dfs graph normal
Arad,Zerind,Oradea,Sibiu,Rimnicu,Craiova,Pitesti,Bucharest
path length: 7
path cost: 762
expanded: 15
visisted(backtracking algorithm): 15
max nodes in memory(backtracking algorithm): 11
```

- عمق اول با عمق محدود درختی:

```
dfs tree limited 5
Arad,Zerind,Arad,Sibiu,Fagaras,Bucharest
path length: 5
path cost: 600
expanded: 25
visisted(backtracking algorithm): 25
max nodes in memory(backtracking algorithm): 5
```

- عمق اول با عمق محدود گرافی:

```
dfs graph limited 5
Arad,Zerind,Oradea,Sibiu,Fagaras,Bucharest
path length: 5
path cost: 607
expanded: 8
visisted(backtracking algorithm): 8
max nodes in memory(backtracking algorithm): 5
```

- عمق اول با افزایش تدریجی عمق درختی:

```
dfs tree deepening
Arad,Sibiu,Fagaras,Bucharest
path length: 3
path cost: 450
expanded: 40
visisted(backtracking algorithm): 40
max nodes in memory(backtracking algorithm): 3
```

- عمق اول با افزایش تدریجی عمق گرافی:

```
dfs graph deepening
Arad,Zerind,Oradea,Sibiu,Fagaras,Bucharest
path length: 5
path cost: 607
expanded: 39
visisted(backtracking algorithm): 39
max nodes in memory(backtracking algorithm): 5
```

- هزینه یکنواخت درختی:

```
uniform tree
Arad,Sibiu,Rimnicu,Pitesti,Bucharest
path length: 4
path cost: 418
expanded: 52
visisted: 131
max nodes in memory: 80
```

- هزینه یکنواخت گرافی:

```
unifrom graph
invalid command
uniform graph
Arad,Sibiu,Rimnicu,Pitesti,Bucharest
path length: 4
path cost: 418
expanded: 12
visisted: 30
max nodes in memory: 10
```

- اولین بهترین حریصانه درختی:

```
best-first tree
Arad,Sibiu,Fagaras,Bucharest
path length: 3
path cost: 450
expanded: 3
visisted: 9
max nodes in memory: 7
```

- اولین بهترین حریصانه گرافی:

```
best-first graph
Arad,Sibiu,Fagaras,Bucharest
path length: 3
path cost: 450
expanded: 3
visisted: 9
max nodes in memory: 7
```

- A\* درختی:

```
A* tree
Arad,Sibiu,Rimnicu,Pitesti,Bucharest
path length: 4
path cost: 418
expanded: 5
visisted: 15
max nodes in memory: 11
```

- A\* گرافی:

```
A* graph
Arad,Sibiu,Rimnicu,Pitesti,Bucharest
path length: 4
path cost: 418
expanded: 5
visisted: 15
max nodes in memory: 11
```

سوال دوم(توضیحات کلی):

هنگامی که برنامه اجرا می شود، یک نمونه از کلاس problem ساخته می شود. در این مسئله، کلاس problem گراف را در خودش دارد. که

این گراف درست مثل سوال اول به صورت لیست مجاورت ذخیره می‌شود. سپس در حلقه‌ی بی‌نهایت از کاربر دستور گرفته می‌شود. در مورد الگوریتم‌های سرکردن شبیه سازی شده، و ژنتیک، آرگمان‌هایی مثل محدودیت زمانی (به صورت عدد اعشاری و در مقیاس میلی ثانیه)، احتمال جهش (به عدد طبیعی و درصد)، فاکتور سرعت سرد شدن (به صورت عدد اعشاری)، اندازه جمعیت (به صورت عدد طبیعی) از کاربر گرفته می‌شوند. ضمناً برای تپهنوردی شروع مجدد تصادفی محدودیت تعداد دفعات شروع مجدد گذاشته شده که ۱۰۰ بار است. برای همین تعداد مشاهده شده و گسترش داده شده ها خیلی زیاد است چون ۱۰۰ بار restart می‌کند.

## سوال دوم (الگوریتم ها):

- تپهنوردی ساده :

```
hill normal
solution:
  1
2 0 0 1
  0 2 0
  2 0 1
value: 2
number of expanded: 10
number of visited: 220
```

- تپهنوردی تصادفی:

```
hill stochastic
solution:
  2
2 0 0 2
  0 2 0
  1 0 1
value: 2
number of expanded: 14
number of visited: 946
```

- تپهنوردی اولین انتخاب:

```
hill first-choice
solution:
  0
2 1 2 1
  0 1 0
  2 0 1
value: 2
number of expanded: 9
number of visited: 47
```

- تپهنوردی شروع مجدد تصادفی:

```
hill restart
solution:
  0
2 1 2 1
  1 0 2
  2 2 1
value: 2
number of expanded: 400
number of visited: 8800
```

- سرد کردن تدریجی: محدودیت زمانی ۲ میلی ثانیه و ضریب دمای ۰.۱۰. در خروجی الگوریتم ستون اول دما، ستون دوم شیب (اختلاف

evaluate) و ستون سوم احتمال انتخاب است که برای state هایی که وضعیت بدتری از state کنونی دارند نمایش داده شده. چون state های بهتر که قطعا انتخاب می شوند. همانطور که می بینیم، محدودیت زمانی داده شده کم بوده و الگوریتم به نتیجه‌ی خوبی نرسیده اما با بیشتر کردن آن و سرد تر شدن، الگوریتم نتایج بهتری بدست می آورد.

```
simulated annealing 10 2
30.2115 0 100
13.6612 -6 64
7.98722 0 100
6.48088 0 100
5.14933 -2 67
4.71476 -2 65
solution:
  2
2 0 1 1
 1 2 0
 1 1 1
value: 10
number of expanded: 14
number of visited: 15
```

- ژنتیک: این الگوریتم با محدودیت زمانی ۸۰۰ میلی ثانیه، احتمال جهش ۱۰٪ و اندازه جمعیت ۱۰۰۰ اجرا شده است. در خروجی ستون اول مینیمم شایستگی در جمعیت، ستون دوم ماکسیمم شایستگی در نسل و ستون سوم میانگین شایستگی در نسل است. الگوریتم ۱۶ نسل اجرا شده.

```
amir@amir-UX310UQ: ~/cpp_workspace/AI_search_algorithms/local
genetic 800 10 1000
11 39 27.38
9 39 27.88
15 39 27.598
13 39 27.646
1 39 27.56
11 39 27.604
7 39 27.642
11 39 27.396
11 39 27.352
9 39 27.464
7 39 27.474
13 39 27.634
11 39 27.764
9 39 27.794
7 39 27.794
11 39 27.548
solution:
  1
1 0 2 0
 0 1 0
 2 0 2
value: 2
generations: 16
```