

بنام خدا



دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش تمرین برنامه نویسی ۱

محمدرضا صمدی

۹۵۳۱۰۵۲

## • سوال اول : مسیریابی شهر ها

نتایج حاصل از اجرا الگوریتم های زیر برای گراف شهر ها به صورت زیر خواهد بود:

### • سطح اول

```
Algorithm Name: BFS Search Algorithm
Visited states number: 9
Expanded states number: 5
Best path: [sibiu, fagaras, bucharest]
Best path cost: 450
Max memory usage (states number): 8
Final state: [State bucharest]
```

### • عمق اول (عمق محدود، عمق نا محدود، افزایش تدریجی عمق)

```
Algorithm Name: DFS Search Algorithm
Visited states number: 11
Expanded states number: 8
Best path: [timisoara, lugoj, mehadia, dobreata, craiova, pitesti, bucharest]
Best path cost: 733
Max memory usage (states number): 10
Final state: [State bucharest]
```

```
Algorithm Name: DLS Search Algorithm
Visited states number: 13
Expanded states number: 9
Best path: [sibiu, rimnicu_vilcea, pitesti, bucharest]
Best path cost: 419
Max memory usage (states number): 11
Final state: [State bucharest]
```

### • هزینه یکنواخت

```
Algorithm Name: UniformCost Search Algorithm
Visited states number: 13
Expanded states number: 12
Best path: [sibiu, fagaras, bucharest]
Best path cost: 450
Max memory usage (states number): 13
Final state: [State bucharest]
```

## • اول بهترین حریصانه

```
Algorithm Name: BestFirstSearchGreedy Search Algorithm
Visited states number: 8
Expanded states number: 3
Best path: [sibiu, fagaras, bucharest]
Best path cost: 450
Max memory usage (states number): 8
Final state: [State bucharest]
```

## • A\*

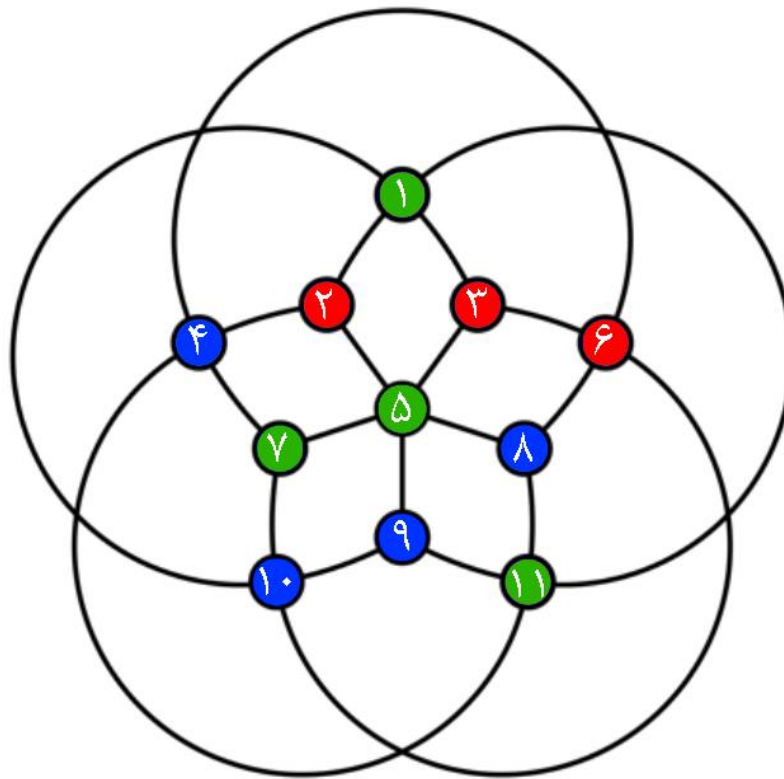
```
Algorithm Name: A* Search Algorithm
Visited states number: 10
Expanded states number: 5
Best path: [sibiu, rimnicu_vilcea, pitesti, bucharest]
Best path cost: 419
Max memory usage (states number): 10
Final state: [State bucharest]
```

## • سوال دوم : رنگ آمیزی گراف

تابع ارزیاب ما تعداد رنگ های متفاوت بین هر دو راس گراف است به عبارت دیگر تعداد یال هایی که دو راس مجاور آن ها رنگ های متفاوتی دارند. هر چه این مقدار بیشتر شود ما هدف خود نزدیک تر میشویم.

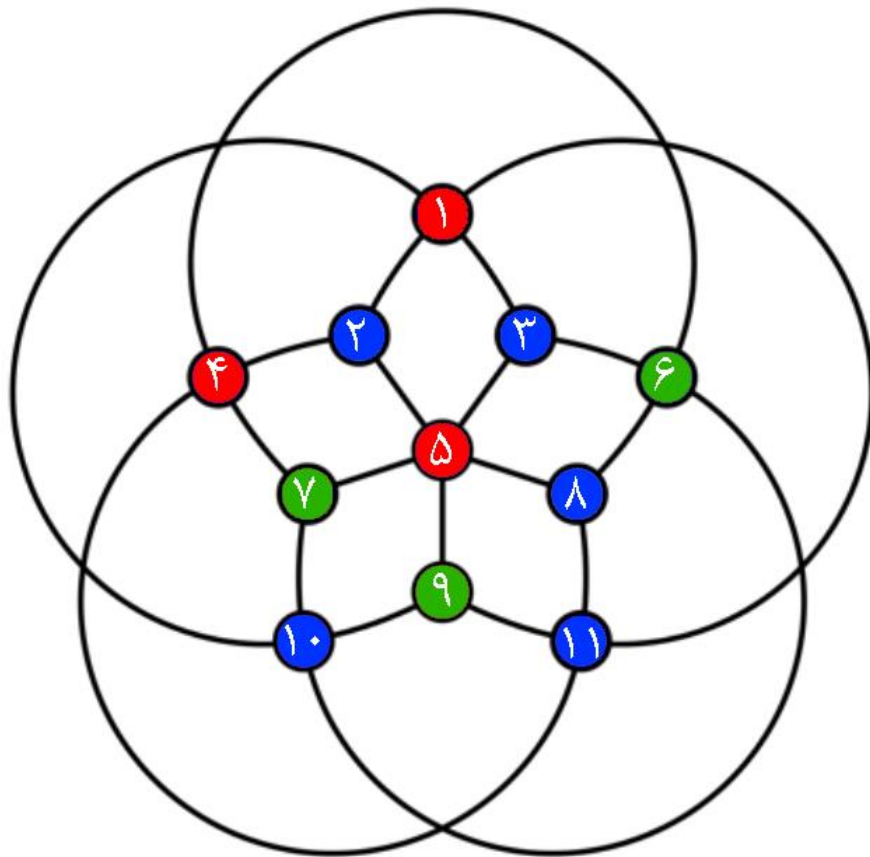
گراف را ما به صورت لیست دنباله دار ذخیره میکنیم به این صورت که هر راس گراف کلاسی به نام Node هستند که هر راس لیستی از همسایه های خود دارد. خروجی های الگوریتم های زیر برای این مسئله به شکل زیر است :

## • تپه نوردی اول بهترین



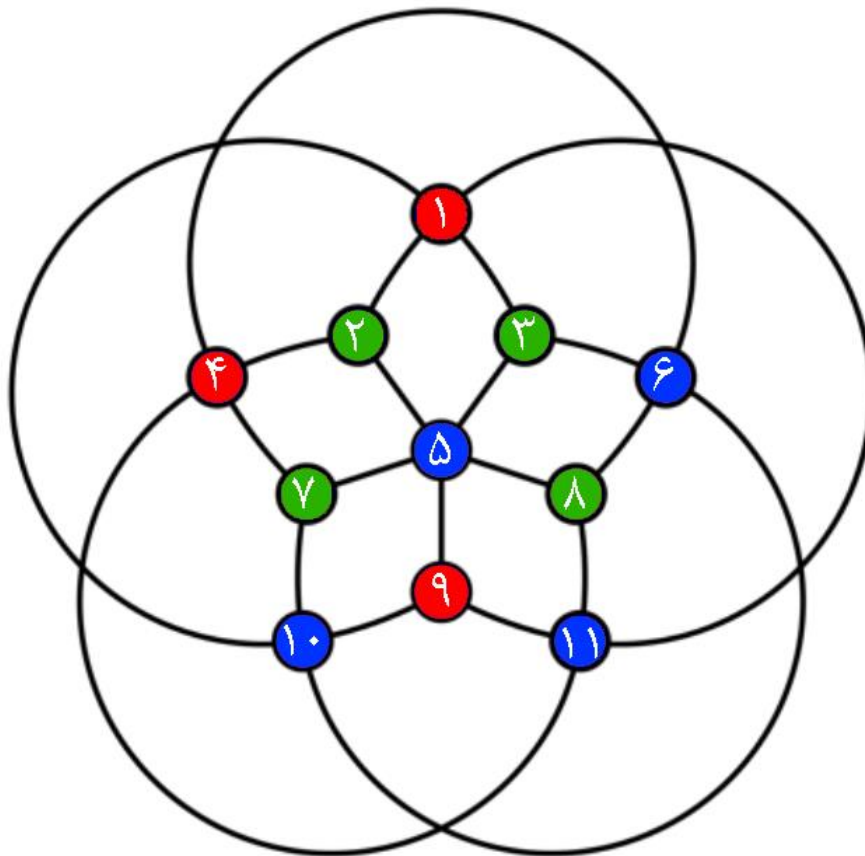
```
Visited State Size : 24
Expanded State Size : 8
Best State is : [
[ G R R B G R G B B B G ] ]
Score = 80.0
```

• تپه نوردی ساده



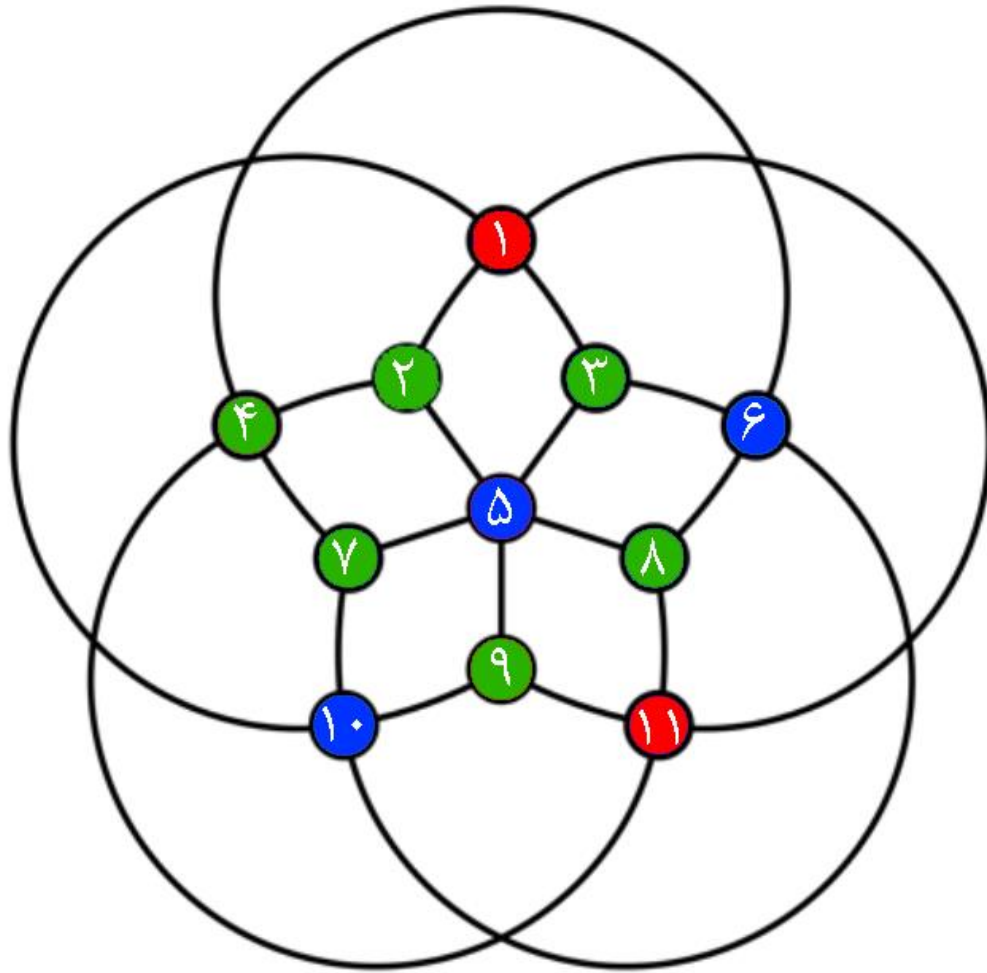
```
Visited State Size : 154
Expanded State Size : 7
Best State is : [
[ R B B R R G G B G B B ] ]
Score = 95.0
```

• تپه نوردی تصادفی



```
Visited State Size : 198
Expanded State Size : 9
Best State is : [
[ R G G R B B G G R B B ]]
Score = 95.0
```

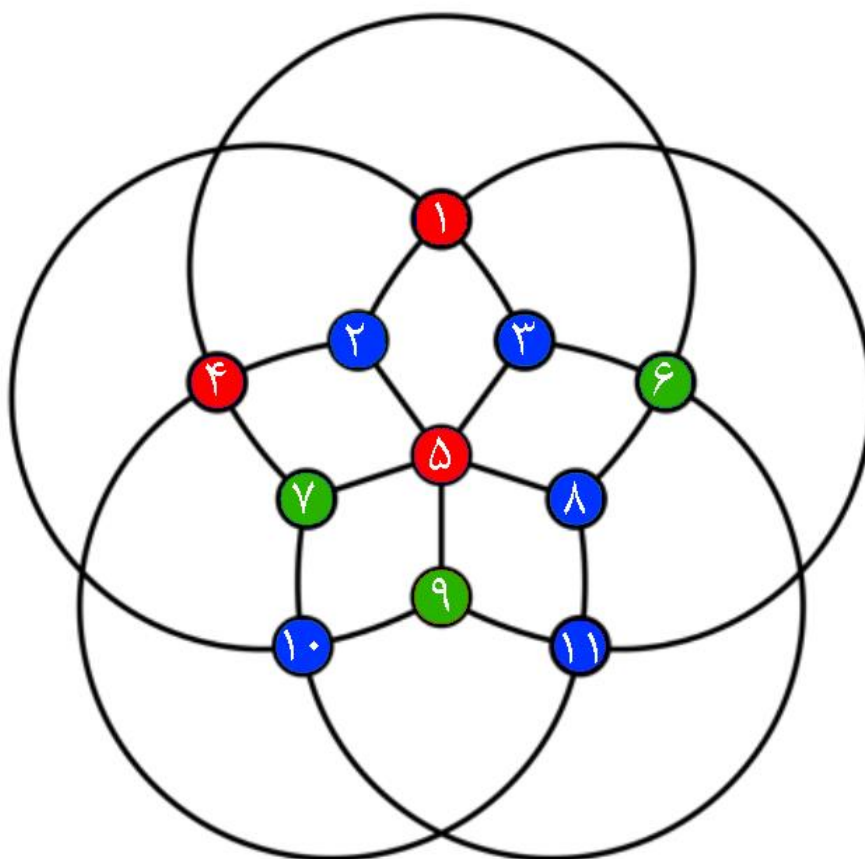
• گرم و سرد کردن تدریجی



```
Visited State Size : 26  
Expanded State Size : 2  
Best State is : [  
[ R G G G B B G G G B R ]]  
Score = 80.0
```



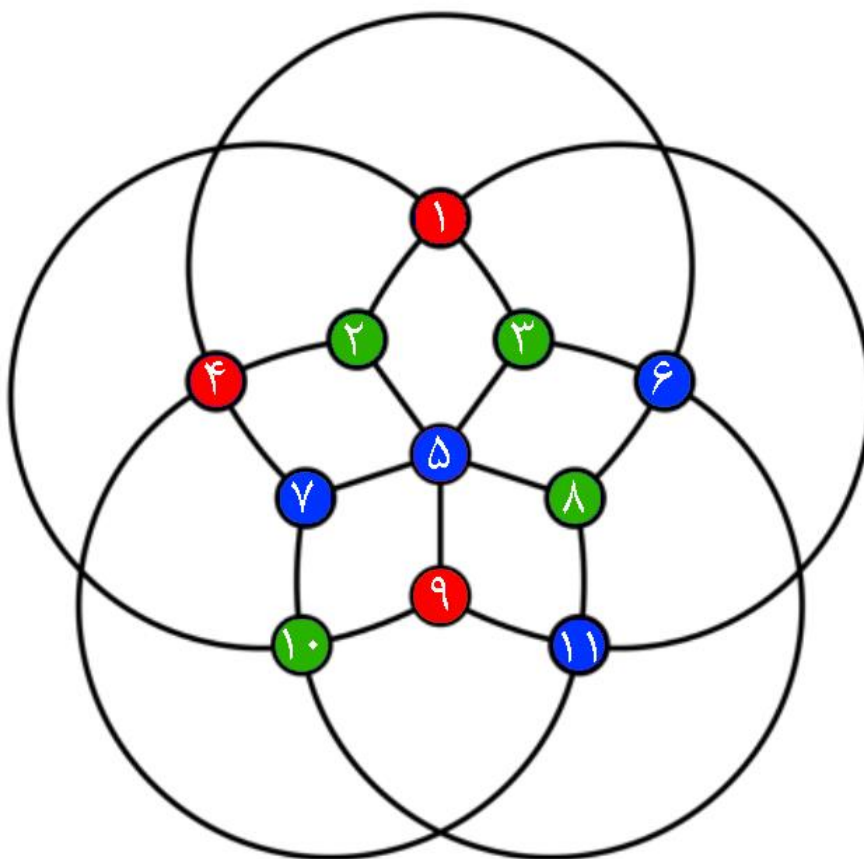
- تپه نوردی با شروع مجدد



```
Visited State Size : 154
Expanded State Size : 7
Best State is : [
[ R B B R R G G B G B B ]]
Score = 95.0
```



## • الگوریتم ژنتیکی



**Max Generate Number = 50**

**Population Number = 40**

**Crossover Rate = 0.4**

**Mutation Rate = 0.2**

**Best Average Fitness = 94.995**

همچنین نمودارهای بهترین و بدترین و متوسط شایستگی در هر نسل به صورت زیر خواهد بود :



- هرچه احتمال جهش بالاتر برود الگوریتم بیشتر به حالت تصادفی نزدیک تر می شود. درواقع جهش در الگوریتم ژنتیک برای پیدایش کروموزوم های جدید استفاده می شود و این باعث می شود که در ناحیه محلی گیر نکنیم. پس با افزایش احتمال جهش کروموزوم های بیشتری تولید میشوند اما باعث می شود همگرایی ما به جواب مطلوب کاهش پیدا کند و اگر جهش ما کم باشد احتمال قرار گیری در یک حالت بهینه محلی بالاتر می رود و خیلی زود همگرا می شود.
- برش یا crossover به منظور همگرایی به حالت مطلوب از جمعیت فعلی استفاده می شود. در واقع هرچه به پایان جستجو نزدیک تر می شویم به برش های بیشتر نیاز داریم. هرچه برش بیشتر شود همگرایی زودتر رخ می دهد و با کم شدن برش همگرایی دیر تر اتفاق می افتد.
- هرچه جمعیت بیشتر باشد در اکثر مواقع باعث بهتر شدن نتیجه و همگرایی زودتر به بهینه سراسری می شود و هرچه مقدار جمعیت کم باشد این همگرایی ممکن است دیر تر اتفاق بیفتد.