

نکات کلی :

-در RandomRestart برای اینکه در صورت سوال حالت بهینه ای تعریف نشده است مجبور شدیم یک متغیر از کاربر بگیریم به عنوان تعداد `iterate` ها که به همان مقدار اجرا شود.

-همه الگوریتم ها به شیوه بازگشتی نوشته شده اند.

مساله رنگ آمیزی گراف (

توابع استفاده شده در این مساله :

`getNeighbors` که تمام رنگ های ممکن را برای هر گراف امتحان میکند و `state` همسایه فقط در رنگ یک نود با `state` فعلی متمایز است.

`getEvaluation` در این تابع جمع مقدار نود های مجاور و غیر هم رنگ را برشمردیم.

```

enter number of nodes:
4
enter number of edges:
5
enter edges ( a-->b ) :
1 2
2 3
3 4
4 1
1 3
Enter MaxColors :
2

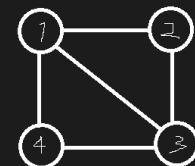
----Solution by simple HillClimbing----
Visited Nodes: 12
Expanded Nodes 3
Evaluation : 4.0
+-----+-----+
| Node | color|
+-----+-----+
| # 1 | 2 |
| # 2 | 1 |
| # 3 | 2 |
| # 4 | 1 |
+-----+-----+

----Solution by RandomRestart HillClimbing----
Expanded Nodes 1000
Evaluation : 3.0

----Solution by Stochastic HillClimbing----
Visited Nodes: 8
Expanded Nodes 1
Evaluation : 4.0
+-----+-----+
| Node | color|
+-----+-----+
| # 1 | 2 |
| # 2 | 1 |
| # 3 | 2 |
| # 4 | 1 |
+-----+-----+

Expanded Nodes 6
Evaluation : 3.0
+-----+-----+
| Node | color|
+-----+-----+
| # 1 | 2 |
| # 2 | 2 |
| # 3 | 1 |
| # 4 | 1 |
+-----+-----+

```



همانطور که مشاهده میشود بهینه ترین جواب را random restart میدهد زیرا با افزایش iterate limit احتمال گیر کردن در لوکال ماکزیمم کمتر میشود.

و غیر دقیق ترین جواب را اول بهترین میدهد.

مساله جدول حروف *

توابع استفاده شده در این مساله :

getNeighbors که دو حرف را جا به جا میکند (تمام جا به جایی های دو حرف)

getEvaluation تعداد حروف پشت سر هم پیدا شده را میدهد . مثلا اگر در

دیکشنری داشته باشیم talk و در جدول به ta رسیده باشیم eval برابر 2 میشود.

```

Enter table Rows And Column size : n m
4 3
Enter table (with a space between each two characters)
a p t
m l b
k l o
u o c
How many words is in dictionary ? :
3
enter dictionary words |
cool
talk
cat
enter Temperature and CoolingRate :
1000
0.003
-----Solved By Simulated Annealing Algorithm-----
Evaluation : 2
VisitedNodes : 151800
ExpandedStates : 2300
a p l
m t b
k l o
u o c
----Solution by RandomRestart HC :-----
eval :11.0
c o o
a t l
l u p
k m b

```

در simulated annealing هر بار اجرا دما را ضرب در (1 - coolingRate) می‌کنیم تا زمانی که مقدار آن به 0.1 برسد (شرط خاتمه) که coolingRate همان نرخ سرد کردن است که کاربر وارد می‌کند.

این الگوریتم همانگونه که میبینید نسبت به Random Restart عملکرد ضعیف تری از خود نشان میدهد.

مساله 3 : کیبورد سازی)

```
Enter initial population count :
10
chromosome 0:      wmeiqrdabhjfs      |      kloynpxvzgute
chromosome 1:      yfgzuhrwbvpqt      |      ksоеixncdmlja
chromosome 2:      yatmondufcbzw      |      iphegrsqvljka
chromosome 3:      tjrqdapzofiyl      |      bwnmxhkgescuv
chromosome 4:      oqnvmhkacbept      |      jxzfuglwdiyr
chromosome 5:      mrlighjzpduox      |      ftakywsnvcbec
chromosome 6:      czjropqthixvu      |      efysnawmgdbki
chromosome 7:      pesrjignbdwta      |      fokmzuhyvlqxe
chromosome 8:      xdhgvepyroasj      |      wbtcuimkqlznd
chromosome 9:      lihxwgnydreot      |      vzqmkbajcusfj
enter population:
5
enter iterationLimit:
1000
enter cuts count in each crossover:
1
enter mutation rate:
0.03
-----Solution of Genetic-----
tjrqdapzofiyl      |      bwnmxhkgescuv      eval : 40.0
```

توابع استفاده شده در این مساله :

`getNeighbors` که دو حرف را جا به جا میکند (تمام جا به جایی های دو حرف)

`getEvaluation` تعداد حروف کاربر سمت چپ را از تعداد حروف سمت راست کم میکنیم منفی قدر مطلق حاصل را جمع میکنیم با تعداد دو حرف های پرکاربردی که در دو قسمت کیبورد هستند ضربدر 2.

اگر کیبورد داده شده دارای حروف تکراری باشد (بدلیل crossover) -1 میدهیم.

نتیجه در 6 نسل :

```
Enter initial population count :
10
chromosome 0:      zdcxhaoypsigl      |      twvumjqefrbkn      eval : 40.0
chromosome 1:      xhnwbtcsaprov      |      miqykegzdljfu      eval : 28.0
chromosome 2:      xoqpufhrhszbet      |      jknagdcylimvw      eval : 28.0
chromosome 3:      pmvbsnoiqljgwd      |      ctfkuelrzaxyh      eval : 30.0
chromosome 4:      qpixtylszncva      |      fbmuhwrdekogj      eval : 30.0
chromosome 5:      ioqfadyrhvgzw      |      ksjncxebutmlp      eval : 40.0
chromosome 6:      wvsnipudhbrfo      |      kqljxameygctz      eval : 34.0
chromosome 7:      lrbiugowpmkst      |      zcdxgjayvfhen      eval : 36.0
chromosome 8:      mvzwoyafqcdbh      |      stielxprkgujn      eval : 36.0
chromosome 9:      wyupjmqogzbih      |      lvscdxfetrkna      eval : 34.0
enter population:
5
enter iterationLimit:
6
enter cuts count in each crossover:
2
enter mutation rate:
0.03
-----Solution of Genetic-----
```

-----Solution of Genetic-----

best Eval: 40.0

average Eval: 40.0

worst Eval: 34.0

best Eval: 40.0

average Eval: 36.0

worst Eval: -1.0

best Eval: 40.0

average Eval: 34.0

worst Eval: -1.0

best Eval: 40.0

average Eval: 30.0

worst Eval: -1.0

best Eval: 40.0

average Eval: -1.0

worst Eval: -1.0

best Eval: 40.0

average Eval: 30.0

worst Eval: 30.0

zdcxhaoypsigl | twvumjqefrbkn eval : 40.0

می بینیم با بیشتر کردن تعداد برش ها evaluation و افزایش نرخ جهش افزایش fitness نداریم...چون ممکن است دو کیبورد کاملاً بی ربط با هم crossover شوند و شاید کاراکتر تکراری بوجود آید و تابع ارزیابی آن -1 شود.

نتایج کیفیت و الگوریتم همگرایی بر جمعیت اندازه تاثیر شایستگی، ارزیابیهای تعداد داشتن نگه ثابت با را بررسی کنید.

با افزایش population افزایش فیتنس را می بینیم .