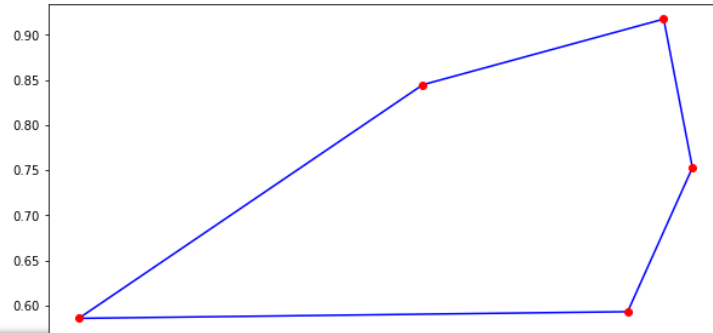


```
fig=plt.figure(figsize=(10,5))
ax1=fig.add_subplot(111)
for first,second in zip(coords[:-1],coords[1:]):
    ax1.plot([first.x,second.x],[first.y,second.y],'b')
ax1.plot([coords[0].x,coords[-1].x],[coords[0].y,coords[-1].y],'b')
for k in coords:
    ax1.plot(k.x,k.y,'ro')
plt.show()
```

```
497 cost= 3.1330665502105366
498 cost= 3.1330665502105366
499 cost= 3.1330665502105366
```



در اکثریم SA ابتدا تعدادی شهر رو به صورت
 تصادفی با مختصات تصادفی و در تقارن فاصله‌های
 تصادفی بر آن آنها ایجاد می‌کنیم.
 مقدار دما رو تعیین و مقداری هم برای ضرب بزرگ‌تر می‌کنیم
 پس برای m بار هزینه بری هزینه و مسرتل خرید و
 می‌کنیم و به تدریج از مقدار دما کم می‌کنیم با ضرب آن در یک ضرب
 و پس دو عنصر با همون دو شهر و با هم جابجایی می‌کنیم
 اگر هزینه و راه‌حل خرید بهتر از قبلی بود خوب اگر قبول
 اگر نه ~~مقدار دما رو کم می‌کنیم~~
 نبود به مقدار تصادفی را با شرط بزرگ‌تر که همون

$$e^{15/t}$$
 است که $\Delta E = \cos D - \cos 1$ چندی می‌کنیم اگر
 مقدار تصادفی کمتر بود می‌پذیریم اگر بیشتر بود روش حل
 و مسرتل را قبول نمی‌کنیم این روند تا بدانشن جواب مورد
 نظر ادامه دادن و در نهایت اگر انشاس بی‌دوم

```
cost0=cost1
else:
    temp=coords[r1]
    coords[r1]=coords[r2]
    coords[r2]=temp

# draw plot
fig=plt.figure(figsize=(10,5))
ax1=fig.add_subplot(111)
for first,seconf in zip(coords[:-1],coords[1:]):
    ax1.plot([first.x,seconf.x],[first.y,seconf.y],'b')
ax1.plot([coords[0].x,coords[-1].x],[coords[0].y,coords[-1].y],'b')
for k in coords:
    ax1.plot(k.x,k.y,'ro')
plt.show()
```

```
inter count of cities5
0 cost= 3.5698334173992317
1 cost= 3.4733394028963573
2 cost= 3.1330665502105366
3 cost= 3.583763464562768
4 cost= 3.473339402896357
5 cost= 3.5168501441604656
6 cost= 3.583763464562768
7 cost= 3.5168501441604656
8 cost= 3.4421947961970076
9 cost= 3.473339402896357
10 cost= 3.538688810699883
11 cost= 3.538688810699883
12 cost= 3.4421947961970076
13 cost= 3.495178069435774
```