

گزارش الگوریتم Hill Climbing

پارسا محمدی

در مسئله TSP، state ما تمام حالات وصل کردن نقاط به هم می باشد. ساکسور ها ترکیب های شهر ها و مسیر هایی است که ما نود هدف می رسم. کاست ما طول مسیر می باشد. و بهترین حالت زمانی است که از تمام شهر ها با کمترین کاست عبور کنیم و به مکان خود بازگردیم.

گزارش کد:

- ابتدا کتابخانه های مورد نیاز را وارد می کنیم.
- بعد دیتا ها مورد نیاز را لود می کنیم. ((توجه شود که این کد در گوگل کلب نوشته شده است و دیتا های ویرایش شده(متن آن حذف شده) در دیرایو من آپلود شده است و من با استفاده از آن ها الگوریتم را نوشتم.))

لینک دیتا قطر در [اینجا](#)

لینک دیتا جیبوتی در [اینجا](#)

لینک کد در [اینجا](#)

من در فایل اصلی دیتا تغییراتی ایجاد کردم و قسمت متن داره فایل را حذف کردم و فقط اعداد باقی ماند تا بتوانم راحت تر آن را پردازش کنم. و فایل کد و دیتا هم در این پوشه موجود است.

- در قسمت بعدی داده ها را نمایش می دهیم تا درک بهتری از داده ها داشته باشیم
- در قسمت بعدی توابع مورد نیاز را نوشته ام
- اولین تابع distance است که فاصله دو نقطه در فضای دو بعدی را حساب می کند. چون داده ها به شکل مختصات هست و ما برای محاسبه کاسب به فاصله بین نقاط نیاز داریم پس نیاز به این تابع داریم.
- تابع بعدی ماتریسی میسازد که فاصله تمام شهر ها از هم در ذخیره شده است. برای اینکه بتوانیم در مراحل بعدی فاصله کاست فانکشن یا همان طول مسیر تا نود هدف را بیابیم بسیار به این ماتریس نیاز داریم.

- تابع بعدی یک جواب رندم را می سازد برای شروع به کار الگوریتم و ما همان جواب اولیه را جواب مسئله در نظر می گیریم تا بعد در طول الگوریتم نقاط بهتر و جواب های بهتر را پیدا کنیم. منظور از جواب یک مسیر از ابتدا ند هدف است که با استفاده از لیستی از ایندکس شهر ها مشخص می شود.

- تابع بعدی لیستی از شهر ها را می گیرد و کاست آن (طول مسیر) را تا نقطه هدف مشخص می کند.

در این تابع درون حلقه `for` محاسبه طول مسیر از ایندکس 1- شروع میشود که این در زبان پایتون به معنا آخرین المان لیست است. اینکار برای این انجام می شود که در مسئله `tsp` نود هدف همان نود آغازین است و باید فروشنده از همه شهر ها یکبار عبود کند و دوباره به همان نقطه آغاز بازگردد پس با طول مسیر بازگشت به نقطه اولیه هم محاسبه بشود.

- تابع بعدی تمام همسایه های موجود (تمام حالت های بعدی) ممکن را برای هم جواب پیدا می کند و روش کار آن هم به این صورت است که جای یکی از المان های لیست را تغییر می دهد. برای مثال : `[1 2 3 4]` می شود `[2 1 3 4]` در یکی از حالات.

- تابع بعدی بهترین حالت بعدی را پیدا می کند. از تمام همسایه های موجود آن همسایه ای که کمترین کاست (طول مسیر) را ند هدف دارد را انتخاب می کند و خروجی می دهد. روش کار آن هم ساده است و فقط کاست همسایه را با کاست فعلی مقایسه می کند.

- تابع `plot_cost` برای رسم نمودار کاست در هر مرحله از الگوریتم تابع کاست را نمایش میدهد. اگر در هر تکرار مقدار کاست ما کاهش یابد این بدین معناست که الگوریتم ما به خوبی کار می کند.

- آخرین تابع تابع تپه نوردی است که قلب کار است. ورودی این تابع ماتریس فواصل بین شهر ها است. در تابع ابتدا یک جواب رندم را با استفاده از تابع `random_solution` که ورودی آن نیز یک ماتریس فواصل است به یک جواب اولیه می رسیم و آن به اسم جواب فعلی می گذاریم. (این جواب لیستی از شهر ها است که فروشنده طی میکند).

در خط بعدی طول مسیر فعلی را محاسبه می کنیم که کاست ما می باشد.

در خط بعد با استفاده از تابع `find_neighbors` تمام همسایه ها (حالت ها بعد پیدا می شود)

و در خط بعدی با استفاده از تابع `find_best_neighbor` بهترین آن ها پیدا می شود. و الگوریتم یک مرحله در جلو تر می رود و جواب مطلوب بعدی به جواب فعلی تغییر می کند.(یک لول از دخت پایین تر می آییم).

در قسمت لوپ هم همین مراحل مدام تکرار می شود تا نود هدف برسیم.

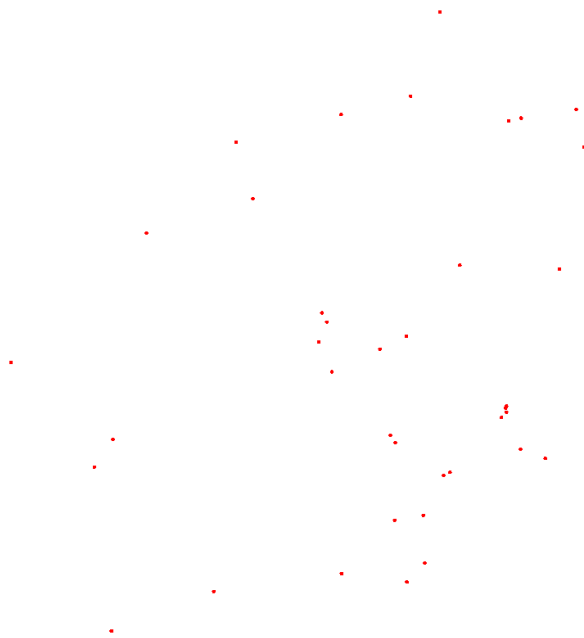
در `main program`

هم توابع اجرا شدند.

و نمودار کاست را رسم می کنیم که ببینیم الگوریتم خوب کار می کند یا خیر.

گزارش عملکرد کد:

جیبوتی:

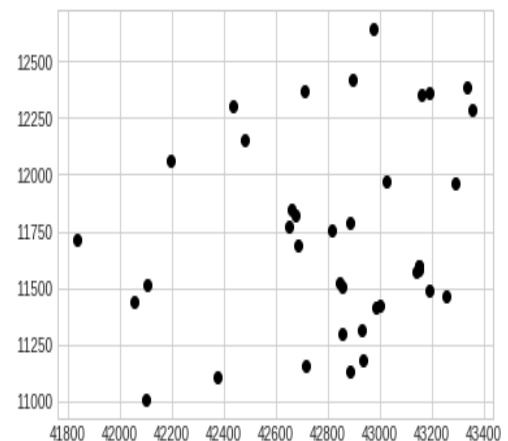


سایت



```
plt.style.use('seaborn-whitegrid')
```

```
plt.plot(y, x, 'o', color='black');
```



کد من

پس دیتا ها به درستی لود شده اند.

```

rent solution is [21, 20, 8, 36, 29, 25, 15, 23, 26, 17, 1, 0, 24, 27, 4, 22, 6, 34, 3, 5, 30, 18, 16, 32, 11, 7, 33, 19, 14, 9, 13, 12, 31, 37, 28, 2, 10, 35] with cost of: 26106.721153800743
rent solution is [37, 34, 12, 24, 13, 16, 23, 31, 6, 35, 30, 14, 18, 25, 2, 7, 26, 4, 5, 0, 21, 3, 22, 10, 36, 28, 29, 20, 1, 9, 8, 15, 11, 32, 17, 27, 19, 33] with cost of: 26106.721153800743
rent solution is [37, 34, 12, 24, 13, 16, 23, 31, 32, 35, 30, 14, 18, 25, 2, 7, 26, 4, 5, 0, 21, 3, 22, 10, 36, 28, 29, 20, 1, 9, 8, 15, 11, 6, 17, 27, 19, 33] with cost of: 23950.075609690244
rent solution is [37, 34, 12, 24, 13, 16, 23, 31, 32, 35, 30, 14, 18, 10, 2, 7, 26, 4, 5, 0, 21, 3, 22, 25, 36, 28, 29, 20, 1, 9, 8, 15, 11, 6, 17, 27, 19, 33] with cost of: 21637.99083078
rent solution is [37, 34, 12, 24, 21, 16, 23, 31, 32, 35, 30, 14, 18, 10, 2, 7, 26, 4, 5, 0, 13, 3, 22, 25, 36, 28, 29, 20, 1, 9, 8, 15, 11, 6, 17, 27, 19, 33] with cost of: 20003.777035460465
rent solution is [37, 34, 12, 24, 21, 16, 23, 31, 32, 35, 30, 26, 18, 10, 2, 7, 14, 4, 5, 0, 13, 3, 22, 25, 36, 28, 29, 20, 1, 9, 8, 15, 11, 6, 17, 27, 19, 33] with cost of: 18418.059263521303
rent solution is [37, 34, 12, 24, 21, 16, 23, 31, 32, 35, 30, 26, 18, 10, 2, 7, 14, 4, 5, 0, 1, 3, 22, 25, 36, 28, 29, 20, 13, 9, 8, 15, 11, 6, 17, 27, 19, 33] with cost of: 16941.749390963058
rent solution is [37, 34, 12, 24, 21, 16, 23, 31, 32, 35, 30, 26, 18, 10, 2, 7, 14, 4, 5, 0, 1, 3, 22, 25, 19, 28, 29, 20, 13, 9, 8, 15, 11, 6, 17, 27, 36, 33] with cost of: 15522.802567507806
rent solution is [37, 34, 31, 24, 21, 16, 23, 12, 32, 35, 30, 26, 18, 10, 2, 7, 14, 4, 5, 0, 1, 3, 22, 25, 19, 28, 29, 20, 13, 9, 8, 15, 11, 6, 17, 27, 36, 33] with cost of: 14370.55139405433
rent solution is [37, 34, 31, 24, 21, 16, 23, 12, 32, 35, 30, 26, 18, 6, 2, 7, 14, 4, 5, 0, 1, 3, 22, 25, 19, 28, 29, 20, 13, 9, 8, 15, 11, 10, 17, 27, 36, 33] with cost of: 13721.46488603425
rent solution is [37, 34, 31, 24, 21, 16, 23, 12, 32, 35, 30, 26, 18, 6, 2, 7, 14, 4, 5, 0, 1, 3, 22, 25, 19, 29, 28, 20, 13, 9, 8, 15, 11, 10, 17, 27, 36, 33] with cost of: 13172.273682985178
rent solution is [37, 34, 31, 24, 21, 16, 23, 12, 32, 35, 30, 26, 18, 6, 2, 7, 8, 4, 5, 0, 1, 3, 22, 25, 19, 29, 28, 20, 13, 9, 14, 15, 11, 10, 17, 27, 36, 33] with cost of: 12785.315203934251
rent solution is [37, 34, 31, 24, 21, 16, 23, 12, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 0, 1, 3, 22, 25, 19, 29, 28, 20, 13, 9, 14, 15, 11, 10, 17, 27, 36, 33] with cost of: 12471.03009432065
rent solution is [37, 34, 31, 24, 21, 16, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 0, 1, 3, 22, 25, 19, 29, 28, 20, 13, 9, 14, 15, 11, 10, 17, 27, 36, 33] with cost of: 12159.941094030377
rent solution is [37, 34, 31, 24, 21, 14, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 0, 1, 3, 22, 25, 19, 29, 28, 20, 13, 9, 16, 15, 11, 10, 17, 27, 36, 33] with cost of: 11859.455484472535
rent solution is [37, 34, 31, 24, 21, 14, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 3, 1, 0, 22, 25, 19, 29, 28, 20, 13, 9, 16, 15, 11, 10, 17, 27, 36, 33] with cost of: 11466.650242874628
rent solution is [37, 34, 31, 24, 21, 14, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 3, 1, 0, 22, 25, 19, 29, 28, 20, 13, 9, 16, 15, 11, 10, 17, 27, 33, 36] with cost of: 11173.236273494109
rent solution is [37, 34, 31, 29, 21, 14, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 3, 1, 0, 22, 25, 19, 24, 28, 20, 13, 9, 16, 15, 11, 10, 17, 27, 33, 36] with cost of: 11010.621058641169
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 3, 1, 0, 22, 25, 19, 24, 28, 20, 13, 9, 16, 15, 11, 10, 17, 27, 33, 37] with cost of: 10862.124935165393
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 3, 1, 0, 19, 25, 22, 24, 28, 20, 13, 9, 16, 15, 11, 10, 17, 27, 33, 37] with cost of: 10730.390074927744
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 32, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 3, 1, 0, 19, 22, 25, 24, 28, 20, 13, 9, 16, 15, 11, 10, 17, 27, 33, 37] with cost of: 10612.843185697178
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 6, 5, 7, 8, 4, 2, 3, 1, 0, 19, 22, 25, 24, 28, 20, 13, 9, 16, 15, 11, 10, 17, 32, 33, 37] with cost of: 10511.917545212984
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 6, 8, 7, 5, 4, 2, 3, 1, 0, 19, 22, 25, 24, 28, 20, 13, 9, 16, 15, 11, 10, 17, 32, 33, 37] with cost of: 10424.466082137687
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 7, 8, 6, 5, 4, 2, 3, 1, 0, 19, 22, 25, 24, 28, 20, 13, 9, 16, 15, 11, 10, 17, 32, 33, 37] with cost of: 10370.88263932078
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 7, 8, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 13, 9, 16, 15, 11, 10, 17, 32, 33, 37] with cost of: 10155.352292624872
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 7, 8, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 16, 15, 11, 10, 17, 32, 33, 37] with cost of: 10112.85406550444
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 8, 7, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 16, 15, 11, 10, 17, 32, 33, 37] with cost of: 10072.513327639013
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 8, 7, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 15, 16, 11, 10, 17, 32, 33, 37] with cost of: 10041.273699400099
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 8, 7, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 15, 10, 11, 16, 17, 32, 33, 37] with cost of: 10033.776053805528
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 15, 7, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 8, 10, 11, 16, 17, 32, 33, 37] with cost of: 10015.172741551822
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 15, 7, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 8, 10, 11, 16, 17, 33, 32, 37] with cost of: 9978.067975794134
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 18, 15, 7, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 8, 11, 10, 16, 17, 33, 32, 37] with cost of: 9974.006903806578
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 17, 15, 7, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 8, 11, 10, 16, 18, 33, 32, 37] with cost of: 9970.699842968941
rent solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 17, 15, 8, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 7, 11, 10, 16, 18, 33, 32, 37] with cost of: 9970.44926457628

```

مراحلی که الگوریتم طی می کند و روند نزولی کاست از تصویر بالا مشخص است.

بهترین جواب:

Best solution is [36, 34, 31, 29, 21, 14, 12, 23, 27, 35, 30, 26, 17, 15, 8, 6, 5, 4, 2, 3, 1, 0, 19, 22, 24, 25, 28, 20, 9, 13, 7, 11, 10, 16, 18, 33, 32, 37] and best cost is 9970.3345802345

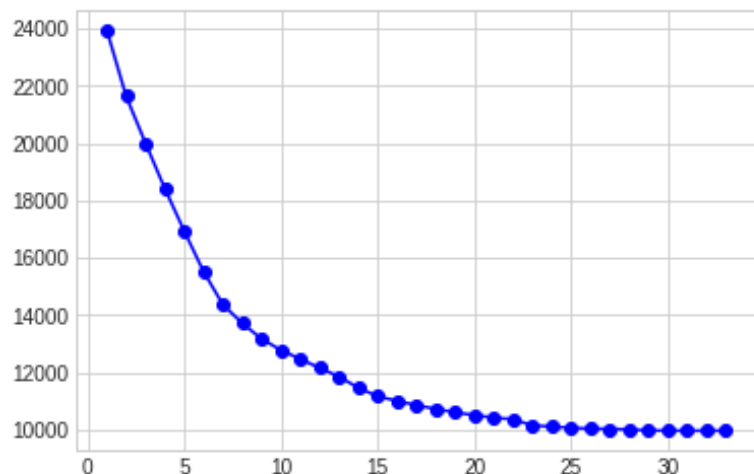
در تصویر بالا شماره مسیر هایی که باید فروشنده طی کند نمایش داده شده است و کمترین کاست 9970 می باشد. لوکال اکستریم است.

و می بینیم که الگوریتم hill_climbing

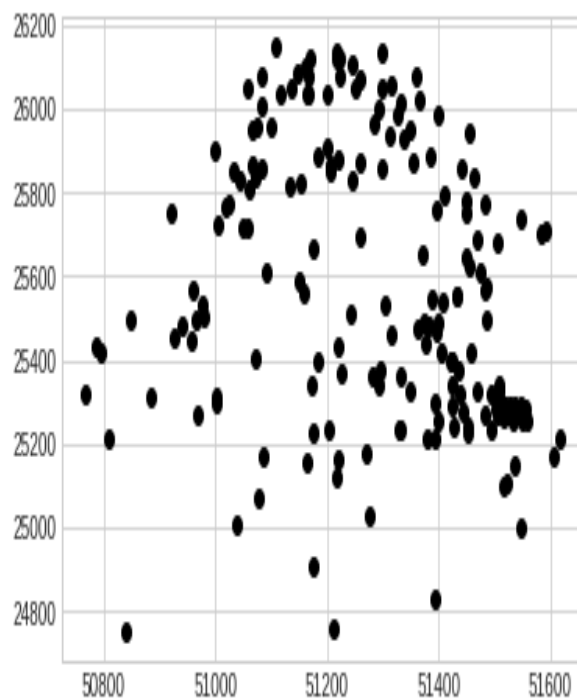
در 33 تکرار نتیجه رو به رو را می دهد.

نمودار عمودی cost

و نمودار افقی تعداد دفعات تکرار الگوریتم است.



قطر:

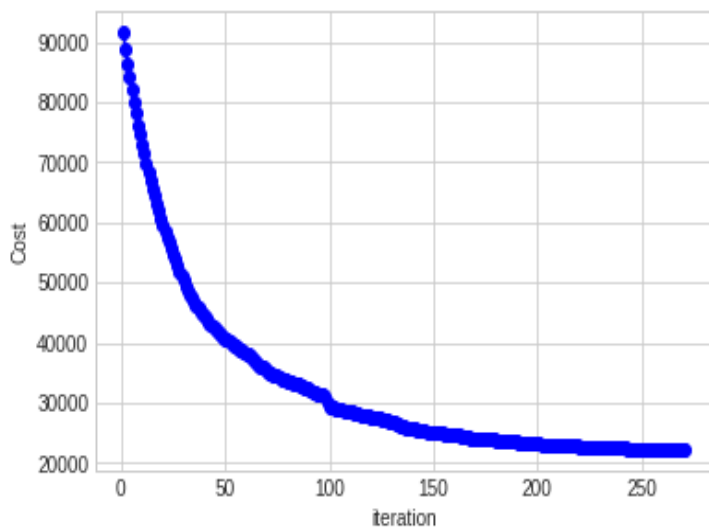


دیتا لود شده در کد من

دیتا در سایت

پس دیتا ها به درستی لود شده اند.

نتایج مربوط به قطر به دلیل زیاد بودن قابل نمایش در این pdf نیست و در فایل به اسم : Qatar Result
Hill Climbing در پوشه hill climbing ذخیره شده است. کمترین کاست 22085.965259686724



نمودار کاست بر حسب هر تکرار قطر دیتا قطر.