گزارش پروژه TSP

1. در روش nearest-neighbor از اولین نقطه شروع میکنیم و بعد به سراغ نزدیک ترین نقطه ی انتخاب نشده میرویم و این روند را تا جایی که همه نقاط انتخاط شوند، ادامه میدهیم.

در متد fill\_array\_by\_file یک آرایه 2 بعدی را با فایلی که به عنوان ورودی میگیریم پر میکنیم.

در متد nearest\_neighbor از اولین نقطه شروع می کنیم و دنبال نقطه بعد با کمک متد find\_next\_nearest\_neighbor هستیم.

اما در روش exhaustive ما باید تمام حالاتی که میتوان نقاط ورودی را پیمایش کرد را بررسی کنیم و کم هزینه ترین حالت را به عنوان خروجی گزارش کنیم.

در متد fill\_array\_by\_file یک آرایه 2 بعدی را با فایلی که به عنوان ورودی میگیریم پر میکنیم.

در متد exhaustive هر جایگشت از نقاط ورودی را به ترتیب به متد calculate\_path\_cost می دهیم تا هزینه پیمودن مسیر را محاسبه کند و آن را با بقیه جایگشت ها مقایسه می کنیم و در نهایت کمترین را بر میگردانیم.



|  |  |
| --- | --- |
| Exhaustive | Nearest neighbor |
| n \* n! | n2 |



ران تایم های الگوریتم nearest-neighbor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| نمونه چهارم | نمونه سوم | نمونه دوم | نمونه اول |  |
| 348 | 372 | 356 | 379 | ران اول us |
| 511 | 336 | 381 | 355 | ران دوم us |
| 341 | 320 | 361 | 367 | ران سوم us |
| 400 | 342 | 366 | 367 | میانگین |

ران تایم های الگوریتم exhaustive

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| نمونه چهارم | نمونه سوم | نمونه دوم | نمونه اول |  |
| 584 | 442 | 1946 | 1886 | ران اول us |
| 614 | 378 | 1891 | 1918 | ران دوم us |
| 583 | 368 | 1875 | 1852 | ران سوم us |
| 593 | 396 | 1904 | 1885 | میانگین |