



به نام خدا
گزارش پروژه اول
ریحانه حیدری ۹۷۳۱۴۸۳

حداقل فاصله بین نقاط:

در این پروژه قصد داریم حداقل فاصله ی دو نقطه را از بین n نقطه ی ورودی بیابیم.

میدانیم اگر با روش Brute-Force این مسئله را حل کنیم از مرتبه ی n^2 خواهد بود و میخواهیم الگوریتمی ارائه دهیم که با تقسیم و حل پیش رود و مرتبه ی زمانی بهتری از الگوریتم قبلی داشته باشد و سریع تر عمل کند.

با توجه به اینکه در فضای سه بعدی هستیم در ابتدا نیاز است که نقاط را در یکی از ابعاد مرتب کنیم که در اینجا این عمل بر روی x انجام می شود پس از مرتب سازی باید تمام نقاط را به صورت بازگشتی به قسمت های مساوی تقسیم کنیم و ممکن است دو نقطه ای که باهم کمترین فاصله را دارند در تقسیم بندی ما یکی از آنها در سمت راست و دیگری در سمت چپ بیافتد پس برای پیدا کردن آنها لازم است ناحیه ای نواری درنظر بگیریم و پس از مرتب سازی مختصات ها طبق y اگر نزدیک ترین جفت در ناحیه نواری نزدیک تر از سایر نواحی باشد باید این فاصله را درنظر بگیریم.

با استفاده از روش اقلیدسی در فضای سه بعدی فاصله ی نقاط را محاسبه میکنیم.

$$c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}$$

پس از پایان combine آخرین مقدار بازگشتی فاصله ی نزدیک ترین دو نقطه هست.

روش کلی حل:

مرتب کردن نقاط

تقسیم نقاط به دو قسمت

پیدا کردن نزدیک ترین دونقطه به طور بازگشتی در دو زیرآرایه

پیچیدگی زمانی:

$$O(n \log(n)) + O(n) + O(n) + O(n) = O(n \log(n))$$

برای بررسی برنامه از نظر مرتبه زمانی پیچیدگی تابع $O(n \log n)$: sort

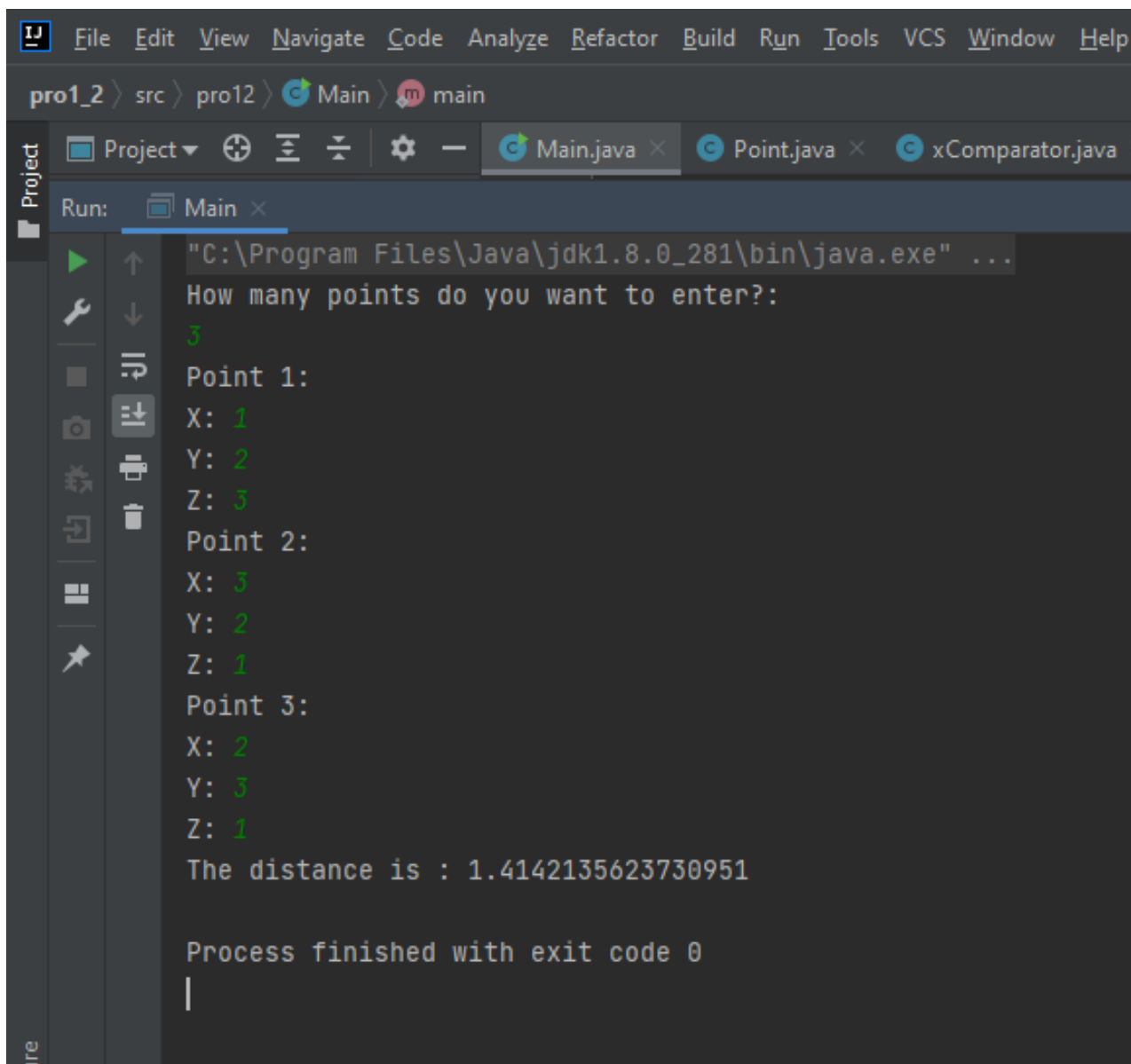
این الگوریتم تمام نقاط را به دو قسمت تقسیم می کند و پس از تقسیم $O(n)$: mid lan aea

در ادامه برای تقسیم نقاط mid lan به دو قسمت: $O(n)$

پیدا کردن نزدیک ترین جفت نقاط در ناحیه $O(n)$: mid lane

"پیدا کردن نوار و تقسیم نوار و پیدا کردن نزدیک ترین جفت در نوار هرکدام مرتبه ی زمانی $O(n)$ را دارند"

در ادامه برای مشاهده عملکرد برنامه دوبار تعدادی نقطه به برنامه میدیم تا خروجی را برای ما محاسبه کند.



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
pro1_2 > src > pro12 > Main > main
Project
Main.java x Point.java x xComparator.java
Run: Main x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
How many points do you want to enter?:
3
Point 1:
X: 1
Y: 2
Z: 3
Point 2:
X: 3
Y: 2
Z: 1
Point 3:
X: 2
Y: 3
Z: 1
The distance is : 1.4142135623730951
Process finished with exit code 0
|
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
src > pro12 > Main > main
Project Main.java Point.java xComparator.java
Main x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
How many points do you want to enter?:
4
Point 1:
X: 100
Y: 1000
Z: 10000
Point 2:
X: 200
Y: 2000
Z: 20000
Point 3:
X: 300
Y: 3000
Z: 30000
Point 4:
X: 400
Y: 4000
Z: 40000
The distance is : 10050.373127401788
Process finished with exit code 0
```