به نام خداوند جان و خرد

نام و نام خانوادگی:محمد حسین کریمی

شماره دانشجویی:9731833

پروژه اول: حداقل فاصله بین نقاط

اسفند99

شرح الگوریتم استفاده شده

با توجه به اینکه فضای سه بعدی داریم ابتدا بر اساس یک بعد سورت می کنیم که در برنامه بر اساس xسورت شده است سپس بعد از مرتب سازی باید تمام نقاط را به صورت بازگشتی به قسمت های مساوی تقسیم کنیم اما این کار کافی نیست چون ممکن است دو نقطه ای که باهم کمترین فاصله را دارند یکی در قسمت راست و دیگری در قسمت چپ.برای پیدا کردن این نقاط یک خط فرضی در نظر می گیریم سپس همه نقاط را با نقطه وسط ارایه چک می کنیم که اگر کوچک تر بود ان را به عنوان کوچک ترین مقدار در نظر می گیریم و جایگزین نقطه وسط می شود این کار را تا زمانی انجام می دهیم که طول آرایه بیشتر از 2باشد

روند برنامه:

در کلاس mainابتدا یک vectorمی سازیم که ساختمان داده ای برای نگه داشتن نقاط هستند به اسم v می سازیم و سپس ورودی از کاربر می گیریم و نقاط گرفته شده را ابتدا در یک آبجکت از کلاس pointریخته و سپس ان را در vاضافه می کنیم سپس دو آبجکت vectorبه نام vecX,vecYبه اندازه طول اصلی درست می کنیم به این دلیل که می خواهیم عملیات سورت را ابتدا روی پارامتر xو سپس روی پارامتر y انجام دهیم.سپس آرایه های ذخیره شده در سه vectorرا به ارایه تبدیل می کنیم به این دلیل که می خواهیم در فراخوانی های بعدی از ان استفاده کنیم.

کلاس utilدر تابع closestکه یک آرایه از جنس pointمی گیرد و تعداد نقاط رو هم می گیرد.سپس یک متغیر به نام closestDistدرست کرده که برای مقدار دهی اولیه آن فاصله عنصر صفرم و اول را در ان می ریزیم

سپس در دو forفاصله نقاط را دو به دو محاسبه می کنیم یعنی هر نقطه با نقطه بعدی ان که در واقع اینجا همان brute forceپیاده شده است(لازم به ذکر است اگر طول آرایه کمتر از 3 بود از همان brute force مراحل را پیش می بریم. در متد divide بر خلاف متد closestکه همه نقاط را چک می کرد در این متد بر اساس الگوریتم ذکر شده در بالا استفاده می کنیم.سپس در متد Mid_Closestتوی اون نواحی فاصله را چک میکنه و کمترین مقدار را بر می گردونه

پیچیدگی:

پیچیدگی زمانی تابع sortبرابرO(nlogn) است این الگوریتم تمام نقاط را به دو قسمت تقسیم می کند.پس از تقسیم عنصر میانی و ناحیه فرضی را بدست میاریم که مرتبه زمانی آن O(n)است در ادامه برای تقسیم نقاط به دو قسمت مساوی مرتبه زمانی آن O(n)لازم است پیدا کردن نزدیک ترین نقطه در ناحیه فرضی از مرتبه O(n)است زیرا محدود به زمان خطی است در نتیجه پیچیدگی زمانی نهای برنامه به صورت زیر است.

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n) + O(n) + O(n)$$
$$T(n) = T(n \log n)$$

```
Enter number of points:
4
Enter three values seperated by space:
1 2 5
Enter three values seperated by space:
6 5 3
Enter three values seperated by space:
2 4 8
Enter three values seperated by space:
1 2 8
End of input
The distance is: 1.7320508075688772

Process finished with exit code 0
```

```
Enter number of points:
6
Enter three values seperated by space:
2 60 52
Enter three values seperated by space:
32 45 21
Enter three values seperated by space:
3 65 48
Enter three values seperated by space:
2 41 53
Enter three values seperated by space:
65 42 58
Enter three values seperated by space:
45 12 35
End of input
The distance is: 3.1622776601683795

Process finished with exit code 0
```