

پروژه دو : Closest Pair of Points

یکی از روش های حل این مسئله ، divide and conquer است. در ابتدا نقاط بر حسب مقدار x شان مرتب میشوند و سپس در هر مرحله نقاط به دو بخش مساوی تقسیم میشوند و کمترین مقدار پیدا شده از سمت چپ را dl و کمترین مقدار پیدا شده از سمت راست را dr مینامیم. $d = \min(dl, dr)$ را به صورت d تعریف میکنیم. زوج نقاطی که یکی از آن ها در سمت چپ و یکی در سمت راست قرار میگیرند نیز باید بررسی شوند اما نیازی نیست که تمام این نقاط بررسی شوند. تنها نقاطی که فاصله آن ها از وسط کمتر از d میباشد نیازمند بررسی میباشند زیرا فاصله نقاطی که در این حد فاصل نیستند نمیتواند از d کمتر باشد. در هر حالت دو بعدی برای هر نقطه که در این حد فاصل قرار دارد حداکثر با ۶ نقطه از طرف مقابلش بررسی میشود. اما برای این کار نیازمند ارایه مرتب شده نقاط بر حسب y نیز میباشیم که اگر در هر مرحله بخواهیم نقاط را مرتب کنیم این کار با پیچیدگی $O(n \log n)$ انجام میشود. اما راه بهتر آن است که یک ارایه مرتب شده از نقاط بر حسب y هم نگه داری کنیم و در هر مرحله که ارایه نصف میشود از ارایه مرتب شده y ها تنها نقاطی که x شان در آن بازه هست انتخاب میشود. بنابراین پیچیدگی این عملیات به $O(n)$ کاهش داده میشود.

در هر حالت سه بعدی نیز روش کار به صورت بالا میباشد با این تفاوتی که برای بررسی نقاطی که در فاصله d قرار میگیرند یکبار مسئله را در حالت دو بعدی برای این نقاط حل میکنیم. به عبارتی رابطه بازگشتی این الگوریتم به صورت زیر میشود :

$$T(n, d) = 2T\left(\frac{n}{2}, d\right) + T(n, d - 1) + O(n)$$

که از حل این رابطه ی بازگشتی به رابطه ی زیر میرسیم :

$$T(n) = O(n(\log n)^2)$$

نمونه مثال اجرا شده توسط برنامه :

```
Enter number of points : 5
Enter points coordinates :
3 -5 4
5 6 -3
7 -10 2
4 10 7
-1 -8 9
Minimum distance is : 7.0710678118654755
```