

توضیح کد: این کد در ابتدا تعداد گره ها را گرفته و سپس مقادیر X گره ها و سپس مقادیر Y گره ها را دریافت می کند. سپس بیشترین و کمترین مقادیر X ها و Y ها را بدست می آوریم. و در خروجی مختصات مستطیل نشان داده می شود.

نمونه خروجی:

```
Please enter number of nodes :
10
X :
-1 3 5 4 9 0 1 -3 5 9
Y :
1 2 4 0 -5 4 9 1 11 12
A: ( -3 , -5 )
B: ( -3 , 12 )
C: ( 9 , 12 )
D: ( 9 , -5 )

Process finished with exit code 0
```

توضیح الگوریتم: برای پیدا کردن کمترین و بیشترین مقدار در یک آرایه، از روش Divide & Conquer استفاده می کنیم، بدین صورت که در هر مرحله هر آرایه را به دو بخش تقسیم می کنیم و این مرحله را بر روی زیر آرایه چپ و راست آرایه اصلی انجام می دهیم تا آرایه به آرایه ای با دو داده (در برخی اوقات به آرایه با یک داده) می رسیم. سپس دو عدد را مقایسه کرده و عدد بزرگتر را به عنوان بیشترین مقدار و عدد کوچکتر را به عنوان کوچکترین مقدار این آرایه به صورت یک آرایه با دو داده که اولی بیشترین مقدار و دومی کمترین مقدار است، برمی گردانیم (برای زیر آرایه با یک داده هر دو مقدار بیشترین و کمترین را برابر با خود عدد قرار می دهیم). سپس در هر آرایه بیشترین مقدار زیر آرایه سمت چپ و راست را با هم مقایسه کرده و آن را به عنوان بیشترین مقدار آرایه اصلی برمی گردانیم. برای کمترین مقدار هم در همان مرحله به همان صورت انجام می دهیم. طبق این روش، ابتدا آرایه تقسیم بر 2 می شود تا به مقدار 1 یا 2 برسد و در این مرحله حداکثر تا 1 مقایسه انجام می دهد ($n/2$ مقایسه). سپس از این مرحله به بعد در هر بار برگشت از آرایه، دو مقایسه (یکی برای بیشترین مقدار و یکی برای کمترین مقدار) صورت می گیرد ($2*(n/4)$). و این مرحله تا جایی ادامه پیدا میکند تا به دو مقایسه برسیم ($2*(n/n)$). در اینصورت مقدار مقایسه برابر است با:

$$n/2 + 2*(n/4) + 2*(n/8) + \dots + 2*(n/n) = 2*n*(1/2 + 1/4 + \dots + 1/n) - n/2$$

$$= 2*n*(1) - n/2 \Rightarrow 2n$$

محاسبه مرتبه زمانی: این الگوریتم همانند الگوریتم merge sort است و الگوریتم آن از مرتبه زمانی $O(n \log n)$ است.