



گزارش پروژه تقسیم و حل

درس طراحی الگوریتم

مسئله «کوچکترین جعبه محدودکننده»

مجتبی فیاضی – 9728973

توضیحات کلی

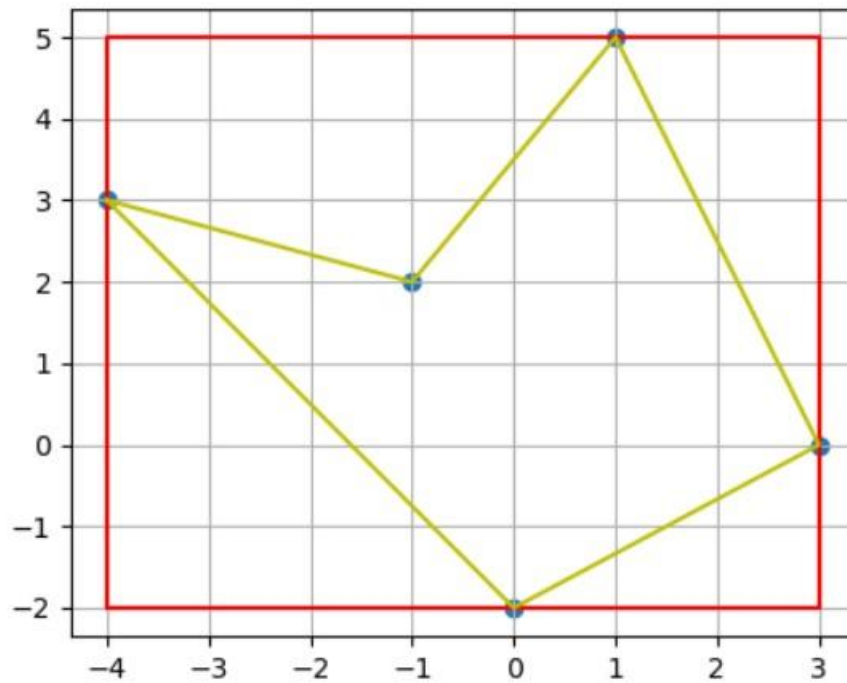
برای پیدا کردن مختصات نقاط گوشه جعبه محدودکننده کافی است ماکزیمم و مینیمم طولها و عرضهای نقاط داده شده را بدست آورد. به این منظور از روش تقسیم و حل برای حل مسئله پیدا کردن ماکزیمم و مینیمم استفاده شده است. روال حل بصورت زیر است:

- اگر طول آرایه مساوی یک بود، همان عنصر را به عنوان ماکزیمم و مینیمم شناسایی می‌شود.
- اگر طول آرایه برابر دو بود، با یک مقایسه ماکزیمم و مینیمم مشخص می‌شود.
- در غیر این صورت، آرایه را به دو قسمت تقسیم می‌کنیم و در هر یک از بخش‌ها بصورت بازگشتی ماکزیمم و مینیمم را مشخص می‌شود. در نهایت بین خروجی‌های بدست آمده دو مقایسه انجام شده و ماکزیمم و مینیمم کل آرایه بدست می‌آید.

```
File: bounding_box.py
1 ~ def findMinMax(i, j, arr):
2 ~
3 ~     if i == j:
4 ~         return (arr[i], arr[i])
5 ~
6 ~     elif j == i + 1:
7 ~         if arr[i] > arr[j]:
8 ~             return (arr[j], arr[i])
9 ~         else:
10 ~             return (arr[i], arr[j])
11 ~
12 ~     else:
13 ~         mid = (i + j)//2
14 ~         min1, max1 = findMinMax(i, mid, arr)
15 ~         min2, max2 = findMinMax(mid + 1, j, arr)
16 ~
17 ~         return(min(min1, min2), max(max1, max2))
18 ~
```

در نهایت برای بدست آوردن مختصات نقاط گوشه جعبه، تابع بالا را یکبار برای طولها و یکبار برای عرضهای نقاط اجرا می‌کنیم.

یک نمونه خروجی برنامه بالا:



مستطیل قرمز، جعبه محدودکننده مد نظر است.

پیچیدگی زمانی

با در نظر گرفتن مقایسه به عنوان عملیات پایه و فرض اینکه n توان 2 باشد، $T(n)$ به صورت زیر در می آید:

$$T(n) = T(n/2) + T(n/2) + 2$$

که در آن:

$$T(1) = 0, T(2) = 1$$

با حل معادله بازگشتی بالا تعداد مقایسه‌ها برابر می شود با:

$$T(n) = 3n/2 - 2$$

بنابراین پیچیدگی زمانی مسئله ما $O(n)$ است.