

Algorithmes et Complexité

TP1 Le plus grand rectangle

Chia-Ling Bragagnolo, Yaakoub Le Guennec

The first file of code (tp1_n4_complexity) is on $O(n^3)$ complexity.

The function contains() :

use a for loop to verify if a point is contained inside a rectangle,
and we use an if condition there to detect if there is any point inside a rectangle,
by comparing the value of x and y position.

if yes, we return True ; if no, we return False

It's complexity is $O(n)$

The function surface() :

for calculating the surface of a rectangle.

It's complexity is $O(1)$

The function currentBasesFrom() :

using a lambda function to substitute x from each point given. Since we go through n points so it's complexity is $O(n)$

Finally the main function solveProblem() :

We substitute x position of the currentBase by using the x position of all the points since we are calculating the surface base on $x = 0$. We always compare the points on the right side of currentBase points and check first if there is any point in the middle of the rectangle with an if condition before comparing the size of surfaces.

It has 3 for nested loops, therefore the complexity is $O(n^3) + O(n) + O(n) + O(1) = O(n^3)$.

The second file of code(tp1_n4_complexity) is also on $O(n^3)$ complexity, but there is a small change in the end.

The function contains() :

use a for loop to verify if a point is contained inside a rectangle,
and we use an if condition there to detect if there is any point inside a rectangle,
by comparing the value of x and y position.
if yes, we return True ; if no, we return False
It's complexity is $O(n)$

The function surface() :

for calculating the surface of a rectangle.
It's complexity is $O(1)$

The function currentBasesFrom() :

using a lambda function to substitute x from each point given. Since we go through n points so it's complexity is $O(n)$

Finally the main function solveProblem() :

We substitute x position of the currentBase by using the x position of all the points since we are calculating the surface base on $x = 0$. We always compare the points on the right side of currentBase points. However in this version we calculate the surface first and find the max surface then verify if this surface contains any points
Therefore in this version the complexity is $O(n^3) + O(n)$ the worse and $O(n^3)$ the best case, even though in either case the Complexity is $O(n^3)$