BENHAJRIA Mouez

Xiaofan XU

TP n°2 : Diviser pour régner

Question 4 :

\*\*\*\*\*\*\*\*\*\*

Initialisation du tableau de deux dimension à 0 .

Charger tous les points .

Parcourir tous les case de tableau avec in teste {

Si en trouve 1 en sauvegarde

}

Parcourir tous le tableau de haut en bas {

Si en trouve un différence entre le pixel [i] et le pixel[i+1] en va tracé un ligne entre le deux .

}

Parcourir tous le tableau dans le sens inverse de gauche a droite {

Si en trouve un différence entre le pixel[i] et le pixel [i+1] en va tracé un ligne entre le deux

}

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Résultat de parcourir de tableau de gauche à droit =>

Résulta de parcourir de tableau de haut en bas =>

Question 5 :

\*\*\*\*\*\*\*\*\*\*

pseudo-code

2 list en param l1 et l2

list <tuple (int,int)> polyline

it2 = iterateur sur L2

it1 = iterateur sur L1

pour chaque point p1 de l1 et p1 de la l2

tant que it1(x)<it2(x)

derniery= it1(y)

it1 suivant

fin tant que

si it1(x) < it2(x) {

si (it1(y) > it2(y)) {

save = it1(x) , it2(x)

}

si (it1(y) < it2(y)) {

save = it1(x), it1(y), it2(y), it2(y\_suivant)

}

Si (it1(y) == it2(y)) {

Save = it1(x), it1(y), it2(y)

}

Si (it1(x) > it2(x) {

si (it1(y) > it2(y)) {

save = it2(x), it1(y)

}

si (it1(y) < it2(y)) {

save = it2(x),it2(y), it2(y\_suivant),it1(y\_suivant)

si (it1(y) == it2(y)) {

save = it2(x), it2(y), it1(y)

}

Fin si ..

}donc l'algorithme est O(n)

Question 6 : voir le fichier ‚‘‘Ligne\_toits “,“ Point‘‘ .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Question 7 : on n'a pas encore fait .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*