

Chercher le plus grand rectangle

Instructions	Explications
Pos[] est un tableau de -1 à 256 PosNb est un entier	Pos[] contient les points et leur indices avant tri PosNb est le nombre de points (maximum 256)
Procédure Search j, xo, dx, dy, s sont des entiers	Chercher le plus grand rectangle <u>Déclarations locales.</u> Début de scan : j indice du point initial, Xo abscisse de départ dx, dy, S : longueur, hauteur et surface courantes
Sous-procédure Next(i, Ytop, Ybtm) Ybtm - Ytop → dy Sortir si (dy ≤ dymin) ou (i ≥ PosNb) i + 1 → i Tant que i < PosNb faire Quitter la boucle si (Pos[i].Y > Ytop) et (Pos[i].Y < Ybtm) i + 1 → i Fin_boucle	Étape élémentaire de scan de gauche à droite dy est la hauteur du secteur en analyse (Ybtm pour bottom) Sortir de Next si espace trop fin ou après le dernier point <u>Chercher le point suivant.</u> Tant qu'il reste des points à examiner On cherche un point obstacle, c'est-à-dire dans le canal entre ces deux ordonnées Ybtm et Ytop Sinon on passe au point suivant
Pos[i].X - Xo → dx dx * dy → s	<u>Ce rectangle est il plus grand que le rectangle maxi actuel ?</u> dx = abscisse du point actuel - abscisse du début du scan Calcul de la surface S
Si (S > rect_maxi.S) et (dy < dx * Ratio) faire Xo → rect_maxi.x, Ytop → rect_maxi.Y dx → rect_maxi.dx, dy → rect_maxi S → rect_maxi.S Fin_si	<u>Remplace le rectangle maxi actuel si plus grand et de ratio OK</u> Enregistrer le coin supérieur gauche Enregistrer la longueur et la hauteur Enregistrer la surface
Si i < PosNb faire Next(i, Ytop, Pos[i].Y) Next(i, Pos[i].Y, Ybtm) Fin si	<u>Passer aux 2 étapes suivantes si elles existent.</u> Traiter depuis le point i entre Ytop et Pos[i].Y Traiter depuis le point i entre Pos[i].Y et Ybtm
Fin de la sous-procédure	
Reset(rectangle maxi) 0 → Pos[-1].X Ly - 1 → Pos[-1].Y Lx-1 → Pos[PosNb].X Ly - 1 → Pos[PosNb].Y	<u>Début effectif de la procédure Search</u> Initialiser à 0 les caractéristiques du rectangle maxi Chien de garde : bord gauche (indice -1) Chien de garde : bord droit (indice PosNb)
De j = -1 jusqu'à PosNb-1 faire Pos[j].X → xo Next(j, 0, Ly-1) Fin de la boucle	<u>En avant !</u> Lancer les différents scans : du bord gauche au dernier point Abscisse du début du scan courant C'est parti pour ce scan ! Point courant j et pleine hauteur.
Fin de la procédure Search	That's all folks !