

PROJET ALGORITHMIQUE AVANCE

1. A. Un cercle est défini par les coordonnées x et y de son centre et son rayon R. Ecrire une fonction qui lit les informations de deux cercles et vérifie s'ils sont en intersection. B. Quel est la complexité de votre algorithme

Solution

A)

Structure Cercle

Entier coordonneeX

Entier coordonneeY

Entier Rayon

Fin Struct

Fonction IntersectionCercle () : Entier

Cercle C[2]

//On crée un Tableau C de type cercle

Entier SommeRayon

Réel Distance

Debut

Ecrire ("Saisir Coordonnée X du premier cercle")

Lire C[0].CoordonneeX *// X1*

Ecrire ("Saisir Coordonnée Y du premier cercle")

Lire C[0].CoordonneeY *//Y1*

Ecrire ("Saisir Coordonnée Rayon du premier cercle")

Lire C[0].Rayon *//R1*

Ecrire ("Saisir Coordonnée X du deuxième cercle ")

Lire C[1].CoordonneeX *//X2*

Ecrire ("Saisir Coordonnée Y du deuxième cercle")

Lire C[1].CoordonneeY *//Y2*

Ecrire ("Saisir Coordonnée Rayon du deuxième cercle") *//R2*

Lire C[1].Rayon

Distance = SQRT((C[1].CoordonneeX - C[0].CoordonneeX)^2 +
C[1].CoordonneeY - C[0].CoordonneeY)^2)

*// la distance entre deux points est donnée par la formule : $d = \sqrt{(X2-X1)^2 + (Y2-Y1)^2}$ on
//calcule la distance entre les deux centres des cercles*

SommeRayon = C[0].Rayon + C[1].Rayon *// La somme des deux rayons*

Si Distance > SommeRayon alors

 Retourne 0

*//si la distance entre les deux centre des
//cercles est supérieure à la somme des*

 Sinon

//deux rayon alors on retourne 0 pour

 Retourne 1

//dire que les cercles ne se croise pas

 Finsi

Fin

*//sinon si la distance entre les deux
//centre des cercles est inferieure ou*

Fin IntersectionCercle

//égale à la somme des deux rayon

//alors on retourne 1 parce que les

//deux cercles se croisent.

B) La complexité est de 23

C'est une complexité constance $\mathcal{O}(1)$

2.A Ecrire une fonction qui lit deux tableaux triés et le regroupe en un seul tableau trié

Solution

Fonction FusionTableau (Entier Tab1[n], Entier n, Entier Tab2[m], Entier m) : **Entier**

Entier t, i, j, tampon, CompteurTab1, CompteurTab2

// la fonction prend 4

Entier Tab3[m + n]

//paramètres, les deux

//tableaux Tab 1 et Tab 2 ainsi

Début

//que la taille des deux

//tableaux respectivement n et m

t = n + m *// t est la taille du tableau fusionné*

i = 0

CompteurTab1 = 1 *//compteur de tab 1*

CompteurTab2 = 0 *//Compteur de tab 2*

Pour i de 0 à (t - 1) faire

 Si CompteurTab1 <= n alors

 Tab3[i] = Tab1[i]

```
CompteurTab1 = CompteurTab1 + 1           //on affecte les valeurs du tableau 1 et
Sinon                                       //du tableau 2 dans le tableau 3 qui
Tab3[i] = Tab2[CompteurTab2]              //est le tableau de fusion
CompteurTab2 = CompteurTab2 + 1
Finsi
FinPour
i = 0                                     // on initialise les variables i et j
j = 0
Tampon = 0                               // une variable qui sera utilisé comme tampon
Pour i de 0 à ( t - 1 )
    Pour j de ( i + 1 ) à ( t - 1 )
        Si Tab3[i] > Tab3[j] alors        //on applique un algorithme de trie
            Tampon = Tab3[i]              // par sélection pour trier le tableau
            Tab3[i] = tab3[j]              // fusionné
            Tab3[j] = Tampon
        Finsi
    FinPour
FinPour
Retourne Tab3[]                           //on retourne le tableau fusionné et trié
Fin
Fin FusionTableau
```

2.B

$$\text{Complexité} = 4 + 4t + 3 + (t * (t-1) * 5) + 1 = 5t^2 - t + 8$$

Complexité Quadratique $\mathcal{O}(t^2)$