

Algorithmique

Correction du DS2

IUT Informatique de Lens, 1ère Année
Université d'Artois

Frédéric Koriche
koriche@cril.fr
2011 - Semestre 1

Exercice 1

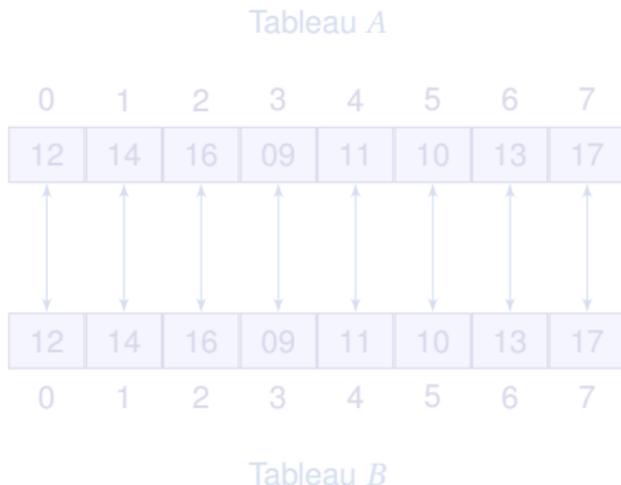
Considérons une agence bancaire contenant 100 comptes en banque. L'agence vérifie si les comptes du jour courant ont évolué par rapport aux comptes du jour précédent.

De manière formelle, le problème est le suivant :

- Données : deux tableaux A et B contenant chacun 100 réels.
- Résultat : "vrai" si les tableaux sont identiques et "faux" sinon.

Exercice 1 (suite)

Rappelons que deux tableaux sont identiques s'ils contiennent les mêmes valeurs aux mêmes positions. Ajouter au programme C++ ci-dessous le code nécessaire permettant de résoudre le problème. Note : on supposera que A et B ont déjà été initialisés (saisis par l'utilisateur).



Exercice 1 (suite)

Rappelons que deux tableaux sont identiques s'ils contiennent les mêmes valeurs aux mêmes positions. Ajouter au programme C++ ci-dessous le code nécessaire permettant de résoudre le problème. Note : on supposera que A et B ont déjà été initialisés (saisis par l'utilisateur).

Tableau A

0	1	2	3	4	5	6	7
12	14	16	09	11	10	13	17

12	14	16	09	11	10	13	17
0	1	2	3	4	5	6	7

Tableau B

Exercice 1 (suite)

Rappelons que deux tableaux sont identiques s'ils contiennent les mêmes valeurs aux mêmes positions. Ajouter au programme C++ ci-dessous le code nécessaire permettant de résoudre le problème. Note : on supposera que A et B ont déjà été initialisés (saisis par l'utilisateur).

Tableau A

0	1	2	3	4	5	6	7
12	14	16	09	11	10	13	17

12	14	16	09	11	10	13	17
----	----	----	----	----	----	----	----

0 1 2 3 4 5 6 7

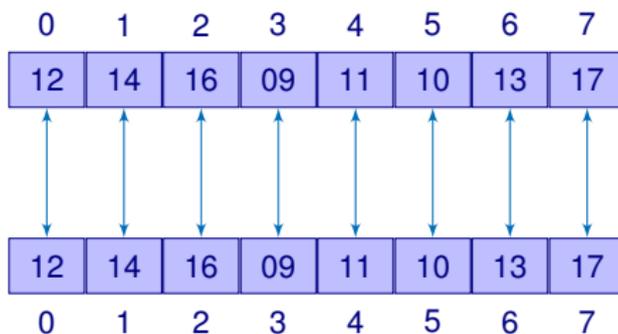


Tableau B

Correction de l'exercice 1

```
#include <iostream>
using namespace std;

typedef double Comptes [100];

int main()
{
    Comptes A,B;
    bool identiques;
    int i;

    i = 0;
    while(identiques && (i < 100))
        {
            identiques = (A[i] == B[i]);
            i++;
        }

    cout << identiques;
    return 0;
}
```

Il s'agit d'une solution parmi plusieurs possibles.

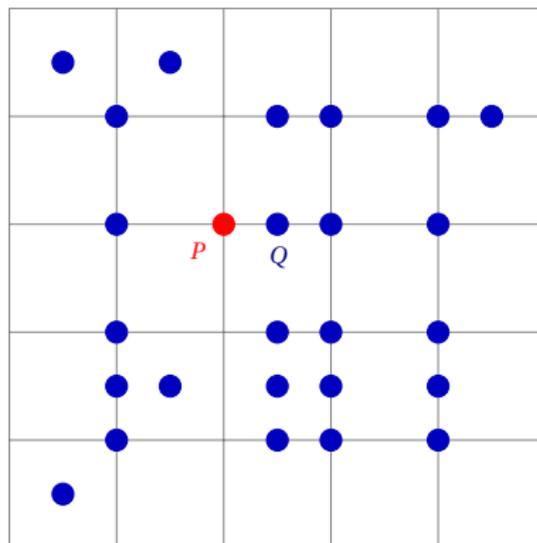
Exercice 2

Dans diverses applications en informatique, nous avons besoin de savoir quel est le plus proche voisin d'un point. De manière formelle, le problème est le suivant :

- Données : un tableau *Tab* de n points dans \mathbb{R}^2 et un point arbitraire P dans \mathbb{R}^2 .
- Résultat : un point Q dans *Tab* tel que Q est un plus proche voisin de P .

Exercice 2 (suite)

Rappelons que Q est un *plus proche voisin* de P si et seulement si la distance (Euclidienne) entre P et Q est plus petite ou égale à la distance entre P et n'importe quel point de Tab .



Le tableau Tab est formé par les points en bleu et le point P est indiqué en rouge

Algorithme 1: plusProcheVoisin

structure Point

réel x
réel y

variables

Point $Tab[n], P, Q$
réel $distanceMinimum, d$
entier i

début

$distanceMinimum \leftarrow +\infty$
pour $i \leftarrow 0$ à $n-1$ **faire**
 $d \leftarrow dist(P, Tab[i])$
 si $d < distanceMinimum$ **alors**
 $distanceMinimum \leftarrow d$
 $Q \leftarrow Tab[i]$
 fin
fin
afficher "Les coordonnées de Q sont : " $Q.x$ ", " $Q.y$

fin

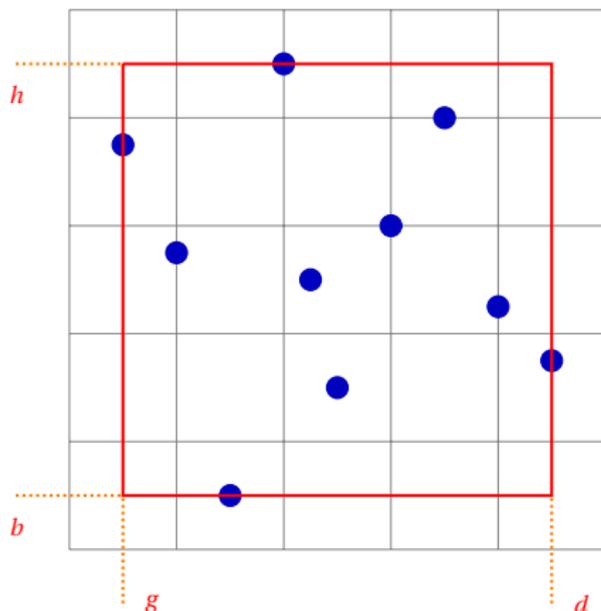
Exercice 3

La notion de rectangle couvrant (*bounding box*) est très commune dans les simulateurs et jeux vidéo pour construire une boîte de collision autour d'un objet constitué par ses points. De manière formelle, le problème est le suivant :

- Données : un tableau *Tab* de n points dans \mathbb{R}^2 .
- Résultat : le rectangle couvrant R de l'ensemble des points de *Tab*.

Exercice 3 (suite)

Rappelons que le rectangle couvrant d'un ensemble de points Tab est le plus petit rectangle qui contient tous les points de Tab . Le rectangle couvrant est représenté par quatre réels : l'abscisse de gauche g , l'abscisse de droite d , l'ordonnée du bas b et l'ordonnée du haut h .



Algorithme 2: rectangleCouvrant (types et variables)

structure Point| réel x | réel y **structure Rectangle**| réel g | réel d | réel b | réel h **variables**| **Point** $Tab[n]$ | **Rectangle** R | **entier** i

Algorithme 3: rectangleCouvrant (corps de la fonction)

début $R.g \leftarrow +\infty$ $R.d \leftarrow -\infty$ $R.b \leftarrow +\infty$ $R.h \leftarrow -\infty$ **pour** $i \leftarrow 0$ à $n-1$ **faire** **si** $Tab[i].x < R.g$ **alors** $R.g \leftarrow Tab[i].x$ **si** $Tab[i].x > R.d$ **alors** $R.d \leftarrow Tab[i].x$ **si** $Tab[i].y < R.b$ **alors** $R.b \leftarrow Tab[i].y$ **si** $Tab[i].y > R.h$ **alors** $R.h \leftarrow Tab[i].y$ **fin****afficher** "Les coordonnées de R sont : " $R.g$ "," $R.d$ "," $R.b$ "," $R.h$ **fin**
