

TP 5 : tableaux et fonctions

Informatique Fondamentale (IF1)

Semaine du 13 Novembre 2006

Comme d'habitude, commencez par créer un répertoire « `tp5` » dans lequel vous travaillerez durant ce TP.

1 Vecteurs

Dans cette partie, on représente le vecteur de coordonnées (x_0, \dots, x_{n-1}) (dans un espace vectoriel de dimension n) par le tableau de taille n dont les éléments sont x_0, \dots, x_{n-1} .

On travaillera `Vect` dans la quelle on écrira les fonctions nécessaires à la resolution des exercices de cette section.

1. Écrivez une fonction

```
public static double produitScalaire(double[] vect1, double[] vect2)
```

qui retourne le produit scalaire des deux vecteurs `vect1` et `vect2` (supposés de même longueur).

Écrivez une fonction `main` qui vous permette de vérifier que le produit scalaire des vecteurs $(2, 3)$ et $(4, 5)$ est bien 26.

2. On rappelle que la norme $|\mathbf{x}|$ d'un vecteur $\mathbf{x} = (x_0, \dots, x_{n-1})$ est définie par

$$|\mathbf{x}| = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{\sum_{i=0}^{n-1} x_i^2}.$$

Écrivez une fonction

```
public static double norme(double[] vect)
```

qui calcule la norme d'un vecteur `vect`. On pourra bien-sûr se servir de la fonction définie à la question précédente.

Écrivez une fonction `main` qui vous permette de vérifier que la norme du vecteur $(2, 3)$ est bien 3,6 environ.

3. Écrivez une fonction

```
public static double[] litVecteurDouble()
```

qui lit un entier n , crée un tableau a de taille n , puis qui lit n nombres réels qu'elle stocke dans le tableau a qu'elle retourne

4. Écrivez une fonction `main` qui lit un vecteur à l'aide de la fonction `litVecteur` puis qui affiche sa norme.

2 Tableaux triés

5. Écrivez une fonction

```
public static int[] litVecteurInt()
```

qui lit un vecteur d'entiers qu'elle retourne. (La programmation par couper/coller est bien-sûr une technique acceptable dans cet exercice.)

6. Écrivez une fonction qui prend comme argument un tableau de nombres et retourne `true` si ce tableau est trié en ordre croissant.

7. Écrivez une fonction `main` qui lit un tableau d'entiers et affiche s'il est trié en ordre croissant.

3 Arguments de ligne de commande

L'argument de la fonction `main`, qu'on appelle `args` lorsqu'on écrit

```
public static void main(String[] args)
```

est un tableau qui contient les arguments du programme. Par exemple, si on exécute le programme `Toto` en tapant

```
$ java Toto bonjour tout le monde
```

la fonction `Toto.main` reçoit en paramètre un tableau à quatre éléments : `"bonjour"`, `"tout"`, `"le"` et `"monde"`.

Écrivez un programme `Echo` qui affiche ses arguments, séparés par des espaces. Par exemple :

```
$ java Echo ala ma kota
ala ma kota
$
```

4 Algorithme d'Euclide

8. On appelle *moyenne harmonique* H d'une famille de nombres (x_0, \dots, x_{n-1}) la valeur

$$H = \frac{1}{\sum_{i=0}^{n-1} \frac{1}{x_i}} \quad \text{si tous les } x_i \text{ sont non-nuls}$$
$$H = 0 \quad \text{sinon}$$

Écrivez un programme qui lit un entier n , puis lit n nombres, puis affiche leur moyenne harmonique.

5 Algorithme d'Euclide

9. L'algorithme dit d'Euclide permet de calculer le pgcd de deux entiers positifs non nuls. Il est basé sur les propriétés suivantes :

- si $a > b$, alors $\text{pgcd}(a, b) = \text{pgcd}(b, r)$, où r est le reste de la division euclidienne de a par b ;
- si $a < b$, alors $\text{pgcd}(a, b) = \text{pgcd}(b, a)$;
- si $a = b$, alors $\text{pgcd}(a, b) = a = b$.

Écrivez une fonction `pgcd` qui calcule le pgcd de ses deux arguments à l'aide de l'algorithme d'Euclide.

6 Recherche

10. Écrivez une fonction

```
int rechercheLineaire(int valeur, int[] a)
```

qui retourne le plus petit i tel que $\mathbf{a}[i] = \mathbf{valeur}$, et -1 si aucun des éléments de \mathbf{a} ne vaut \mathbf{valeur} .

11. Lorsque le tableau \mathbf{a} est trié, il est plus efficace de procéder en utilisant un algorithme appelé *recherche par dichotomie*. En effet, avec la méthode de l'exercice précédent, pour savoir si aucun des éléments du tableau \mathbf{a} n'est égal à \mathbf{valeur} il faut comparer \mathbf{valeur} successivement à tous les éléments du tableau.

La recherche par dichotomie manipule trois entiers p , m et g (pour « petit », « moyen » et « grand »). À tout moment, $\mathbf{a}[p] \leq \mathbf{valeur} \leq \mathbf{a}[g]$.

Initialement, p vaut 0 et g la taille du tableau moins un. À chaque étape de calcul, m vaut $(p + g)/2$; si $\mathbf{a}[m] \leq \mathbf{valeur}$, alors p devient m , sinon c'est g qui devient m . Le calcul s'arrête lorsque $p = m$.

Écrivez une fonction

12. Écrivez une fonction

```
int rechercheDichotomie(int valeur, int[] a)
```

qui utilise une recherche par dichotomie pour trouver un indice i tel que $\mathbf{a}[i] = \mathbf{valeur}$, et retourne -1 si un tel indice n'existe pas.

13. Écrivez un programme qui :

- lit un entier \mathbf{valeur} ;
- lit un tableau d'entiers \mathbf{a} ;
- vérifie si ce tableau est trié;
- recherche un indice i tel que $\mathbf{a}[i] = \mathbf{valeur}$ en utilisant, selon le résultat du point précédent, une recherche linéaire ou une recherche dichotomique;
- affiche cet indice i .

On pourra bien-sûr se servir des fonctions définies aux exercices précédents.