

Structures, recherches et tris

On a vu en cours plusieurs algorithmes de recherche et de tri de données. Les exemples donnés en cours permettait uniquement de rechercher et de trier des entiers. Dans ce TD, on va créer des structures relativement complexes, définir des fonctions permettant de les manipuler facilement, et adapter les algorithmes de recherche et de tris vu en cours afin qu'ils puissent manipuler les structures que nous avons créés.

► Exercice 1. Des belles structures

Soit les types structurés suivants :

- Un point est caractérisé par ses coordonnées cartésiennes.
- Un cercle est caractérisé par son centre (un point) et son rayon.
- Un rectangle est caractérisé par son coin supérieur gauche, sa longueur et sa hauteur.
- Une roue est composée de deux cercles (le pneu et la jante).
- Une voiture vue de côté est composée de deux roues, une partie basse (un rectangle), une partie haute (un rectangle), deux fenêtres (deux rectangles).

Écrire la définition de ces structures.

► Exercice 2. Des belles fonctions

Pour chacune des structures du premier exercice, écrire les fonctions :

- création : qui prends en entrée les paramètres nécessaire à l'initialisation des champs et renvoie une variable initialisée du type correspondant.
- affichage : qui prend en entrée un type structuré et affiche sa valeur.
- comparaison : qui prend en entrée deux types structurés $s1$ et $s2$ et renvoie

$$\begin{cases} \text{une valeur négative} & \text{si } s1 < s2 \\ 0 & \text{si } s1 = s2 \\ \text{une valeur positive} & \text{si } s1 > s2 \end{cases}$$

- échange : qui prend en entrée deux pointeurs de types structurés et échange les valeurs des variables pointées.

Pour les comparaisons :

- les points : on compare leur distance à l'origine.
- les cercles : on compare leur rayon.
- les rectangles : on compare les aires (longueur \times hauteur).
- les roues : on compare les pneus. Si les pneus sont de taille égale, on compare les jantes.
- les voitures : on compare la taille de leur partie basse.

► **Exercice 3. Chercher et trier des voitures**

Écrire une fonction qui :

- prend en entrée un tableau de voitures et renvoie la position dans le tableau de la plus petite d'entre elles.*
- écrire une fonction qui trie un tableau de voitures dans l'ordre croissant.*

Pour chacune de ces fonctions, on se demandera ce qu'il suffirait de changer pour obtenir les mêmes fonctions en remplaçant le type voiture par n'importe quel autre.