

Cours 4 - TD

Exercice 1 - Manipulation de templates

1. Proposez une fonction permettant de trier un tableau d'entiers de taille variable. Vous emploierez un tri *simple* : le tri à bulle.

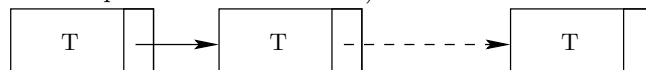
```
début
  trie = faux;
  tant que non trie faire
    trie = vrai;
    pour i ∈ {0, ..., |T| - 1} faire
      si T[i] > T[i + 1] alors
        echanger(T[i], T[i + 1]);
        trie = faux;
fin
```

Algorithme 1: Tri à bulle

2. Surchargez la fonction pour des tableaux de `double`.
3. Faire de cette fonction un template et proposez deux exemples d'utilisation avec un tableau d'entiers et un tableau de `double`.
4. On considère le type `Point` comportant 2 entiers `x` et `y`. Quel(s) opérateur(s) devez-vous surcharger pour le type `Point` afin de pouvoir l'employer dans le cadre du template précédent ? Implémentez ce(s) opérateur(s) et proposez un exemple d'utilisation du template pour trier un tableau de `Point`.
5. Faire du type `Point` un template (on peut souhaiter par exemple que les coordonnées soient exprimées en `int` ou `double`, etc). Faire du(des) opérateur(s) nécessaire(s) à l'emploi du template de tri des templates.

Exercice 2 - Introduction aux algorithmes génériques

1. Rendre le template du tri à bulle indépendant de la notion de tableau en lui passant un pointeur sur le premier élément et un pointeur sur un marqueur de fin (par exemple le pointeur sur le dernier élément + 1). Donner un exemple d'utilisation.
2. On considère un template de liste chaînée simple. Écrire le template correspondant (une structure avec la valeur et le pointeur sur le suivant).



3. Redéfinir un template d'opérateur `++` de façon à pouvoir faire avancer un pointeur sur un élément de la liste chaînée vers son suivant. *Note : le marqueur de fin est le pointeur 0.*
4. Donner un exemple d'utilisation du template de tri sur une liste chaînée.