

Programmation Fonctionnelle

Tris

Le but de ce TP est d'implémenter les différents tris en Haskell.

Exercice 1 : Le tri à bulle

Pour rappel, l'algorithme du tri à bulle est le suivant :

```
tri_a_bulles(tableau T)
  pour i de 0 à (Taille de T) - 1
    pour j allant de 0 à (Taille de T) - 1
      si T[j] > T[j+1]
        échanger(T[j], T[j+1])
```

1. Ecrivez une fonction `tri_a_bulles`.

Exercice 2 : Le tri par insertion

Pour rappel, l'algorithme du tri par insertion est le suivant :

```
tri_insertion(tableau T)
  pour i de 1 à (taille de T) - 1
    x = T[i]
    # décaler vers la droite les éléments de T[0]..T[i - 1] qui sont plus petit que x
    j = i
    tant que j > 0 et T[j - 1] > x
      T[j] = T[j - 1]
      j = j - 1
    T[j] = x
```

2. Ecrivez une fonction `tri_insertion`.

Exercice 3 : Le tri par sélection

Pour rappel, l'algorithme du tri par sélection est le suivant :

```
tri_selection(tableau T)
  pour i de 1 à (taille de T) - 1
    min = i
    pour j de i + 1 à (taille de T)
      si t[j] < t[min]
        min = j
    si min != i
      echanger(t[i], t[min])
```

3. Ecrivez une fonction `tri_selection`.

Exercice 4 : Le tri fusion

Pour rappel, l'algorithme du tri fusion est le suivant :

```
tri_fusion(tableau T)
  T1 = tri_fusion(T[0..(taille T)/2])
  T2 = tri_fusion(T[(taille T)/2..(taille T)])
  retour fusion(T1, T2)
```

4. Ecrivez une fonction `tri_fusion`.

Exercice 5 : Le tri rapide

Pour rappel, l'algorithme du tri rapide est le suivant :

```
tri_rapide(tableau T)
    T1 = tri_rapide(tous les éléments plus petit ou égal à T[0] sans T[0])
    T2 = tri_rapide(tous les éléments plus grand que T[0] sans T[0])
    return T1 + T[0] + T2
```

5. Ecrivez une fonction `tri_rapide`