

TD numéro 4 : tris, récursivité profonde

À préparer avant la séance

1. Tri par sélection du minimum

Le tri par sélection du minimum consiste à sélectionner le minimum de la liste à trier, à le mettre en début de liste, puis à recommencer sur la liste à laquelle on a enlevé ce minimum.

- Définir une fonction qui calcule le minimum d'une liste de nombres.

```
(minimum '(2 8 5 1 6)) → 1
```

- Définir une fonction `enleve` qui supprime un élément d'une liste (on se contentera de supprimer la première occurrence de l'élément).

```
(enleve 5 '(2 8 5 6 4)) → (2 8 6 4)
```

- Définir la fonction `tri-min` qui utilise les deux précédentes pour trier une liste de nombres par sélection du minimum.

```
(tri-min '(8 2 4 3 1 5 9)) → (1 2 3 4 5 8 9)
```

À faire pendant la séance

2. Récursivité profonde

- Écrire une fonction calculant le nombre total d'éléments d'une liste quelconque.

```
(compter '(1 2 (q w e) (r (e (w t)) 3 (e)) (o))) → 12
```

- Écrire une fonction qui prend une liste et ajoute 2 à chaque nombre de cette liste en profondeur.

```
(ajoute2 '(2 b (10 a (56 3) 5) 4)) → (4 b (12 a (58 5) 7) 6)
```

- Écrire une fonction qui, étant donnée une liste contenant des entiers, remplace en profondeur les entiers de la liste par un booléen spécifiant si l'entier correspondant est pair ou non.

```
(pairs '(2 (3) 5 (6 (7)))) → (#t (#f) #f (#t (#f)))
```

3. Tri à bulles

Le tri à bulles est aussi un tri par sélection : on sélectionne l'élément minimum de la liste à trier, on le met en première position, et l'on recommence avec le reste de la liste.

Pour extraire l'élément minimum de la liste, on utilise une technique plus efficace que dans l'exercice 1 : on le fait remonter (comme une bulle) vers le début de la liste par échanges successifs de deux éléments voisins.

- Écrire la fonction `bulle` qui fait remonter le minimum en début de liste.
- Écrire la fonction `tri-bulles`, qui trie une liste de nombres par appels successifs à la fonction `bulle`.
- Dérouler le tri à bulles sur la liste (7 9 1 6 2 3)
- Si `l` est une liste de longueur `n`, quel est le nombre de comparaisons effectuées par l'appel :

```
(tri-bulles l)
```