

TDTP 4 : tris, calculs en remontant/descendant

À préparer avant la séance

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

TD 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 le tri par sélection du minimum : 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.

```
(bulle '(7 9 6 1 4 3)) → (1 7 9 6 3 4)
```

- 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) ?
```

Calculs en remontant ou en descendant

- TD • Définir une fonction qui, étant donnée une liste d'entiers, construit une liste de deux sous-listes : celle contenant les éléments impairs et celle contenant les éléments pairs.

```
(parite '(1 2 6 7 5)) → ((1 7 5) (2 6))
```

Écrire une version où les calculs sont faits en remontant et une autre où les calculs sont faits en descendant.

- TP • Définir une fonction qui étant donnée une liste, construit une liste de deux sous-listes : celle contenant les symboles et celle contenant les nombres.

```
(symb-nb '(tor 1 tue la 2 3 pin 4)) → ((tor tue la pin) (1 2 3 4))
```

Écrire une version où les calculs sont faits en remontant et une autre où les calculs sont faits en descendant.

TP

Tri par insertion

Nous allons trier une liste en utilisant le tri par insertion. Le principe de ce tri consiste à trier récursivement le *cdr* de la liste, puis à insérer le *car* au bon endroit dans la liste.

- Écrire la fonction `insere` qui insère au bon endroit un nombre dans une liste de nombres triée.

`(insere 4 '(1 2 5 7)) → '(1 2 4 5 7)`

- Écrire la fonction `tri-insertion` qui trie une liste de nombres.

`(tri-insertion '(7 4 9 1)) → (1 4 7 9)`

Pour s'entraîner (exercices supplémentaires facultatifs)

- Définir une fonction qui, étant donné un nombre x et une liste de nombres L , construit deux listes : celle des nombres de L inférieurs ou égaux à x , et celle des nombres de L supérieurs à x .

`(separe 3 '(5 1 2 3 6 4)) → ((1 2 3) (5 6 4))`

Écrire une version où les calculs sont faits en remontant et une autre où les calculs sont faits en descendant.