

TP 6 : tris (à faire sur 2 séances)

Ce TP est à déposer à la fin de cette séance sur moodle (<https://foad.univ-rennes1.fr/course/view.php?id=1006723>). Vous pouvez également déposer une version améliorée de votre TP jusqu'à la fin de cette semaine.

Fichier à récupérer depuis moodle : tp6.mlw

Il est inutile d'utiliser des variables fantômes dans ce TP.

Avant d'écrire un invariant de boucle, il est fortement conseillé de dessiner sur une feuille le tableau partiellement trié après un nombre quelconque d'itérations.

Exercice 1 : tri par sélection

1. Spécifier puis programmer le sous-programme `minimum` renvoyant l'indice du plus petit élément d'un sous-tableau `t[m..n]`.

```
let minimum (t : array int) (m n : int) : int (* à compléter *)
```
2. Spécifier puis programmer le sous-programme `tri_selection` de tri par sélection d'un tableau d'entiers.

```
let tri_selection (t : array int) : unit (* à compléter *)
```

Le principe du tri par sélection dans l'ordre croissant d'un tableau `t` dont les éléments sont indicés de 0 à `N-1` est le suivant.

- Rechercher le plus petit élément de `t`, puis l'échanger avec l'élément `t[0]`.
- Rechercher ensuite le 2^e plus petit élément parmi les éléments restants de `t` et l'échanger avec `t[1]`.
- Itérer le procédé jusqu'à ce que le tableau soit trié.

Indications :

- L'échange de deux éléments utilisera le sous-programme `swap` vu au cours 3.

```
val swap (a: array int) (i: int) (j: int) =  
  requires { 0 <= i < length a /\ 0 <= j < length a }  
  ensures { exchange (old a) a i j }
```
- Dans l'invariant de boucle, utiliser le prédicat `(sorted_sub t i j)` spécifiant que le sous-tableau `t[i .. j]` est trié.

Exercice 2 : tri par insertion

Programmer et prouver un programme effectuant un tri par insertion (cf. http://fr.wikipedia.org/wiki/Tri_par_insertion).

```
let tri_insertion (t : array int) : unit (* à compléter *)
```

Exercice 3 : tri à bulles

Programmer et prouver un programme effectuant un tri à bulles.

```
let tri_bulles (t : array int) : unit (* à compléter *)
```
