

# TP d'algorithmique et structures de données

## TP1 – Ensembles

### Exercice 1

Le type « Ensemble » permet les opérations suivantes :

- `size( $E$ )` qui renvoie le nombre d'éléments de l'ensemble  $E$ .
- `is_empty( $E$ )` qui renvoie vrai si l'ensemble  $E$  est vide et faux sinon.
- `add( $E$ ,  $x$ )` qui rajoute l'élément  $x$  à l'ensemble  $E$ .
- `remove( $E$ ,  $x$ )` qui supprime l'élément  $x$  de l'ensemble  $E$ .
- `in( $E$ ,  $x$ )` qui renvoie vrai si l'élément  $x$  est présent dans l'ensemble  $E$ .
- `is_equal( $E_1$ ,  $E_2$ )` qui renvoie vrai si l'ensemble  $E_1$  est égal à l'ensemble  $E_2$  et faux sinon.
- `intersect( $E_1$ ,  $E_2$ )` qui renvoie un nouvel ensemble qui est l'intersection des ensembles  $E_1$  et  $E_2$  (c.-à-d. qui contient les éléments présents dans  $E_1$  et  $E_2$  à la fois).
- `union( $E_1$ ,  $E_2$ )` qui renvoie un nouvel ensemble qui est l'union des ensembles  $E_1$  et  $E_2$  (c.-à-d. qui contient les éléments présents dans  $E_1$  ou dans  $E_2$ ).
- `difference( $E_1$ ,  $E_2$ )` qui renvoie un nouvel ensemble qui est la différence des ensembles  $E_1$  et  $E_2$  (c.-à-d. qui contient les éléments présents dans  $E_1$ , mais pas dans  $E_2$ ).

**À faire :**

1. Écrire une bibliothèque en Javascool permettant de gérer le type « ensemble d'entiers ». Utiliser l'implantation des ensembles à l'aide des tableaux triés.
2. Proposer une séquence de tests utilisant toutes les fonctions ci-dessus.