TP d'algorithmique et structures de données

TP2 – Multiensembles

Exercice 1

Le type abstrait de données *Multiensemble* est une collection non-ordonnée d'objets qui peuvent y être présents en plusieurs exemplaires. Il permet les opérations suivantes :

- `size(`M`)` qui renvoie le nombre d'éléments (total) de l'ensemble M.
- `support(`M`)` qui renvoie l'ensemble support de M (contenant tous les éléments de M).
- `is_empty(`M`)` qui renvoie vrai si le multiensemble M est vide et faux sinon.
- `multiplicity(`M`, `x`)` qui renvoie le nombre d'éléments x dans le multiensemble M.
- `in(`E`, `x`)` qui renvoie vrai si l'élément x (de type T) est présent dans le multiensemble M.
- `add(`M`, `x`)` qui rajoute l'élément x (de type T) au multiensemble M.
- `addm(`M`, `x`, `n`)` qui rajoute n copies de l'élément x (de type T) au multiensemble M.
- `remove(`M`, `x`)` qui supprime l'élément x (de type T) du multiensemble M.
- `removem(`M`, `x`,`n`)` qui supprime n copies de l'élément x (de type T) du multiensemble M.
- `sum(`M_1`, `M_2`)` qui renvoie un nouveau multiensemble qui est la somme des multiensembles M_1 et M_2 (c.-à-d. qui contient les éléments présents dans M_1 et M_2 et dont la multiplicité est la somme des deux multiplicités).
- `difference(`M_1`, `M_2`)` qui renvoie un nouveau multiensemble qui est la différence des multiensembles M_1 et M_2 (c.-à-d. qui contient les éléments présents dans M_1 et éventuellement dans M_2 et dont la multiplicité est la différence des deux multiplicités).

À faire :

1. Écrire une bibliothèque en Javascool permettant de gérer le type « multiensemble d'entiers ». Utiliser impérativement l'implantation efficace à l'aide des tableaux triés (en temps $O(n + m)$ pour les deux dernières opérations, $O(\log n)$ pour les tests et $O(n)$ pour les modifications).
2. Proposer une séquence de tests utilisant toutes les fonctions ci-dessus.