

## EXAMEN DE RATTRAPAGE MELO

1 heure 30

### Exercice 1.- (Tri par seaux, documents permis)

L'idée fondamentale du **tri par seaux** (bucketsort en anglais) est de se procurer un certain nombre de seaux  $S_1, \dots, S_m$ , de parcourir le tableau non trié et de jeter chaque élément dans un seau de façon telle que tous les éléments du seau  $S_i$  soient plus petits que tout élément du seau  $S_j$  si  $i < j$ . Ensuite on trie les éléments de chaque seau et on regroupe tout dans un tableau (qui sera donc trié) contenant d'abord les éléments du seau  $S_1$  dans l'ordre, puis ceux du seau  $S_2$  et ainsi de suite jusqu'au seau  $S_m$ .

La seconde idée est de considérer un grand nombre de seaux contenant peu d'éléments. La façon de trier les éléments de chaque seau n'est donc pas très importante. On pourra utiliser une méthode simple telle que le tri par insertion ou le tri à bulles.

La façon de distribuer les éléments dans chaque seau se fait en choisissant une fonction de hachage.

Cette méthode a surtout un intérêt lorsque l'on sait que les éléments à trier sont, par exemple, des entiers compris entre 0 et un certain entier  $MAX$ .

- 1<sup>o</sup>) Définir une constante  $MAX$  (par exemple 100 000) et une classe `AsmLSeau`, dont les attributs sont une liste d'au plus 16 entiers ainsi qu'un entier `length` spécifiant la dimension effective utilisée et pour méthodes `insert(a)` qui place l'entier  $a$  dans la liste (et qui renvoie faux s'il y a déjà 16 éléments dans le tableau), `sort()`, qui trie la partie utile de la liste (grâce à un tri sélection), `len()` qui renvoie la longueur utile de la liste et `elem(i)` qui renvoie le  $i$ -ième élément de la liste.

- 2<sup>o</sup>) Définir une méthode :

`bucketsort(tab, n)`

qui trie la liste d'entiers naturels `tab` de longueur  $n$  par seaux et renvoie la liste triée.

[ On définira une liste  $S$  de  $MAX/16$  seaux, que l'on initialisera tous à `null`. On parcourera la liste non triée et on placera l'élément en cours dans le

*bon seau : si celui-ci est vide, il faut le créer ; dans tous les cas, il faut insérer l'élément dans le seau. On triera les seaux non vides. On concaténera enfin les seaux non vides pour obtenir la liste triée.*

*L'opérateur + permet de concaténer deux listes. ]*

- 3°) Mettre le corps de la fonction `bucket_sort()` sous forme normale.