

CaML - Licence Pro - 2009-10

TP2: Tris et recherche

5 mars 2010

Question 1 Construire une fonction *appartient* qui détermine si une liste *l* triée par ordre décroissant contient un élément *x*.

Question 2 Questions Additionnelles :

1. *remove* qui retire un élément d'une liste.
2. *isIn* qui vérifie si un élément est dans une liste.
3. *addBegin* qui ajoute un élément en début de liste.
4. *nbOcc* qui calcule le nombre de fois qu'un élément *x* apparaît dans une liste *l*.
5. *addEnd* qui ajoute un élément en fin de liste.
6. Ecrire la fonction concaténation de deux listes *l1* et *l2*.
7. *mirror* qui retourne l'image inverse d'une liste.

Question 3 On considère à présent les listes comme des ensembles. Ecrire les fonctions d'union, d'intersection et de différences de deux ensembles.

Question 4 (Tri par insertion) On écrit ce tri en deux étapes :

1. Ecrire une fonction *insert* qui insère un élément à sa place dans une liste triée par ordre décroissant.
2. Ecrire la fonction *tri_insert* qui réalise le tri par insertion.

Question 5 Tri par fusion

On veut à présent définir le tri fusion en plusieurs étapes. Le problème principal est la manipulation de types complexes : les paires.

1. écrire une fonction *diviser* qui coupe une liste en deux listes de mêmes tailles (... autant que possible).
2. écrire une fonction *fusionner* qui fusionne deux listes
Indications : en OCaml, une paire d'élément s'écrit *(elt1, elt2)*.
3. écrire le tri fusion à partir de ces deux fonctions.
Indications : De plus, vous aurez besoin de définir des variables locales pour les fonctions *diviser* et *tri_fusion*.