

TP numéro 2

1 Tri à bulles

- Définir le prédicat bulle(L1,L2) qui construit la liste L2 qui est la liste L1 dans laquelle l'élément le plus petit est remonté en première place.
- Définir le prédicat tribulle(L1,L2) qui implémente le tri à bulles.

2 Arbres binaires

- Définir un prédicat qui vérifie l'appartenance d'un élément à un arbre. Comment le prédicat se comporte-t-il en génération ? Que se passe-t-il si on inverse l'ordre des clauses ?
- Définir un prédicat qui construit la liste des valeurs des feuilles d'un arbre

3 Arbres binaires de recherche

On considère des arbres binaires de recherche (ABR) dont les étiquettes sont des nombres positifs.

- Définir un prédicat d'insertion d'un élément dans un ABR.
- Définir un prédicat qui calcule le plus grand élément d'un ABR et retourne également l'arbre privé de cet élément.
- Définir un prédicat qui supprime un élément d'un ABR.
- Définir un prédicat qui construit un ABR à partir d'une liste de nombres par des insertions successives.
- Définir un prédicat qui trie une liste en passant par la construction d'un ABR.

4 Ensembles (facultatif)

Rappel : Un ensemble est une collection d'éléments non ordonnée, on ne tient donc pas compte de la place et chaque élément de l'ensemble apparaît une et une fois seulement. On représente les ensembles par des listes.

- Définir un prédicat to_list(X,Y) qui transforme une liste X en un ensemble Y (on élimine toutes les occurrences qui sont présentes plus d'une fois dans la liste)

Exemple : `to_list([a,b,c,b,d,e,d],L).` → `L=[a,b,c,d,e]`

- Définir un prédicat inclus(X,Y) qui teste l'inclusion de l'ensemble X dans l'ensemble Y.

Exemple : `inclus([a,c,v],[e,v,a,c,f]).` → `Yes`

- Définir un prédicat inter(X,Y,Z) qui construit l'ensemble Z intersection des deux ensembles X et Y (une ligne avec findall).

Exemple : `inter([a,b,c,d,e],[d,r,a,e,f],L).` → `L=[a,d,e]`