

TP2: Fonctions récursives, tri et recherche

Fonctions récursives

1. Écrire une fonction 'decimal_a_binaire(n)' qui prend un entier positif 'n' et renvoie une chaîne représentant 'n' en binaire (ex : 13 → "1101"). La fonction doit être récursive.
2. Écrire une fonction récursive 'syracuse(n)' qui retourne la liste des termes de la suite de Syracuse débutant à 'n' jusqu'à 1.
3. Écrire une fonction récursive 'aplatir(liste)' qui transforme une liste contenant des sous-listes en une liste à un seul niveau.
4. Écrire une fonction récursive 'est_equilibre(expr)' qui vérifie si les parenthèses et crochets d'une chaîne sont correctement appariés et imbriqués.

Algorithmes de tri

5. Écrire une fonction 'tri_selection(tab)' qui trie une liste d'entiers en place selon la méthode du tri par sélection.
6. Écrire une fonction 'tri_insertion(tab)' qui trie une liste d'entiers selon le tri par insertion.
7. Écrire une fonction 'fusionner_listes(L1, L2)' qui fusionne deux listes triées d'entiers en une seule liste triée.

Algorithmes de recherche

8. Écrire une fonction 'recherche_lineaire(tab, x)' qui retourne l'indice de la première occurrence de 'x' ou -1 si absent.
9. Écrire une fonction 'recherche_dichotomique(tab, x)' qui effectue une recherche binaire de 'x' dans un tableau trié.
10. Écrire une fonction 'position_insertion(tab, x)' qui retourne la position où insérer 'x' dans une liste triée pour maintenir l'ordre.
11. Écrire une fonction 'deux_sum(tab, S)' qui retourne un tuple de deux valeurs distinctes de 'tab' dont la somme est 'S', ou None si aucune paire n'existe.