

## TP2 PROLOG (2h)

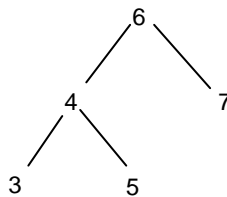
### Exercice n° 1 : tri par insertion

Écrire le prédicat `insérer(X, L, S)` qui insère l'entier  $x$  dans la liste triée  $L$  pour produire la liste triée  $S$ . En déduire le prédicat `triParInsertion(L, S)` selon les deux formes de récursivité.

### Exercice n° 2 : encodage d'arbre binaire de recherche

On considère des arbres binaires dont les nœuds sont étiquetés par des entiers. Les arbres sont des arbres binaires de recherche, c'est-à-dire que si  $n$  est l'étiquette d'un nœud  $N$ ,  $g$  le maximum des nœuds situés dans son fils gauche  $G$  et  $d$  le minimum des nœuds situés dans son fils droit  $D$ , alors on a la relation suivante :  $g \leq n < d$ .

Un nœud  $N$  est représenté par une liste à trois éléments :  $[G, n, D]$  où  $G$  représente le sous arbre gauche de  $N$ ,  $D$  le sous-arbre droit de  $N$  et  $n$  l'étiquette de  $N$ . L'arbre vide est représenté par le symbole  $[]$ . L'arbre suivant :



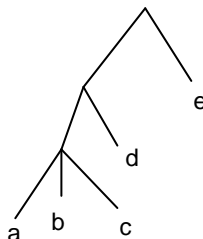
est donc représenté en Prolog de cette manière :

```
[[[], 3, []], 4, [[], 5, []]], 6, [[], 7, []]].
```

Écrire le prédicat `ajout(T, A, R)` qui ajoute l'élément  $A$  dans l'arbre  $T$  pour donner l'arbre  $R$ .

### Exercice n° 3 : encodage d'arbre n-aire à étiquettes sur les feuilles

On travaille cette fois sur des arbres n-aires où seules les feuilles sont étiquetées. L'arbre sera représenté de manière classique. Ainsi, l'arbre :



s'écrit de la manière suivante :

```
[ [ [ a, b, c ], d ], e ].
```

Écrire le prédicat `memes_feuilles(x, y)` qui est vrai si les arbres `x` et `y` ont les mêmes feuilles de gauche à droite.

On utilisera à cette fin un prédicat `aplatir(A, L)` où `L` est la liste des feuilles de l'arbre `A`, de gauche à droite.