

# TMI - 2

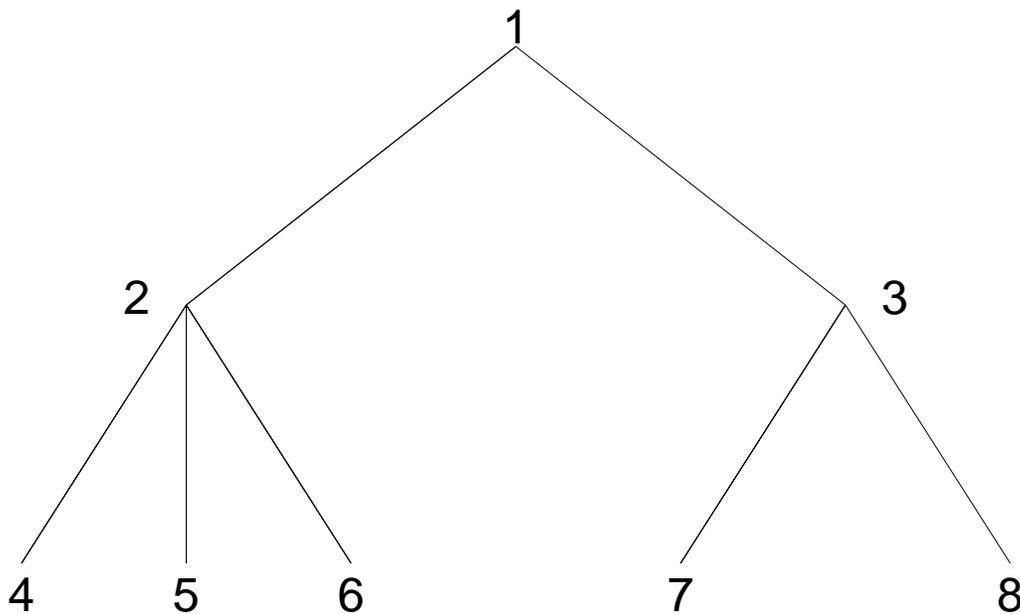
## TP. n° 5

Dans ce TP on va travailler avec les arbres à partir de la représentation par matrice d'adjacence du TP n°4. Puis en utilisant une représentation particulière pour les arbres enracinés comme celle utilisée pour la question 5 du TP n°3.

1. Ecrire une fonction `is_tree(A)` qui prend une matrice d'adjacence en entrée et renvoie `true` ou `false` selon que le graphe (non orienté) passé en paramètre soit ou pas un arbre.



2. Représentation des arbres enracinés : on va représenter les arbres comme :



par la liste `[1, [2, [3], [4], [5]], [3, [6, 7]]]`. Autrement dit `[nom_du_noeud, suite_des_fils]`.

a) Ecrire une fonction qui à partir d'un arbre représenté de cette façon donne sa matrice d'adjacence (en numérotant les sommets dans l'ordre de lecture).

b) Ecrire une fonction à qui l'on passe la matrice d'adjacence d'un arbre et le numéro d'un sommet (future racine), et qui donne l'arbre enraciné résultant (en notation de liste).

c) Ecrire les fonctions `ordre(T)`, `degre(T)`, `profondeur(T)`, `is_binary(T)`, `feuilles(T)`,

racine( $T$ ) où  $T$  est un arbre représenté comme plus haut.

d) Ecrire les fonctions  $\text{Descendance}(T, num)$  et  $\text{Ascendance}(T, num)$  qui donne respectivement l'ensemble des descendants et des ascendants du noeud qui porte le nom  $num$ .

e) Ecrire  $\text{NoeudBinaire}(T, 0001)$  qui renvoie le noeud de l'arbre  $T$  qui a le nom binaire  $0001$  (par exemple).

