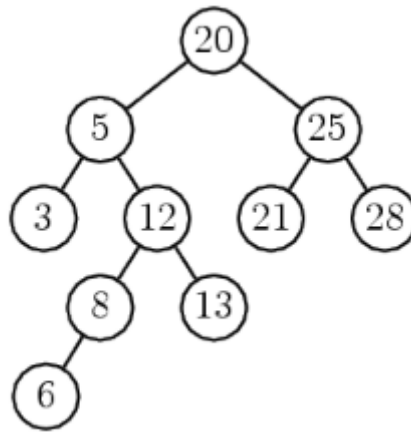


M1 – AL - APP

Algorithmique et Complexité
Arbres et graphes
TD3
Arbre binaire de recherche

LAURIER Alexis

Exercice 1 :



- 1°) Cet arbre est-il un arbre binaire de recherche ?
- 2°) Insérer les nœuds avec l'étiquette : 50, 5, 25, 19. Certains nœuds posent-ils des soucis ?
- 3°) Effectuer le parcours infixe de l'arbre ainsi obtenu. Que remarquez-vous ?
- 4°) Implémenter une classe permettant d'instancier l'arbre ci-dessus, d'y insérer des éléments et d'en faire le parcours infixe.

Exercice 2 :

- 1°) À l'aide d'un arbre binaire, trie la liste de valeurs suivante : 25 60 35 10 5 20 65 45 70 40 50 55 30 15. Pour cela, construisez un arbre binaire de recherche puis effectuez un parcours infixe de ce dernier.
- 2°) Implémenter une méthode sur la classe précédemment créée permettant de créer un arbre binaire de recherche à partir d'une liste quelconque de valeurs pouvant être comparées. Tester votre classe à l'aide d'une liste aléatoire de 10000 valeurs entières générées aléatoirement.
- 3°) Comparer le temps de recherche de 100 valeurs différentes à l'intérieur d'une liste de 10000 valeurs entières générées aléatoirement. Vous comparez une méthode basée sur un arbre binaire de recherche et une méthode classique de parcours d'une liste d'éléments.
- 4°) Comparer le temps de tri d'une liste de 10000 valeurs entières générées aléatoirement entre la méthode `sort` de Collections et votre classe basée sur un arbre binaire de recherche. Notez les différences en faisant varier le nombre de valeurs de la liste d'entrée. Expliquez les résultats obtenus.