Algorithmique et Structures de Données
TP 3 - Les arbres

Halim Djerroud (hdd@ai.univ-paris8.fr)

Partie I

Pour les trois quatres parties (Arbre binaire, Arbre binaire de recherche, Parcours des arbres et les tas) vous devez rédiger un document (LaTeX de préférence) permettant d'expliquer le fonctionnement de chaque partie. Vous pouvez vous aider le l'outil `graphviz` pour générer les grahiques.

Arbre binaire :

1. Analyser l'implémentation donnée.

Arbre binaire de recherche :

1. Analyser l'implémentation donnée.

Parcours des arbres :

1. Parcours en largeur : Analyser l'implémentation donnée.
2. Parcours en profondeur : Analyser l'implémentation donnée.

Les tas :

1. Analyser l'implémentation donnée.

Arbre binaire équilibrés (AVL) :

En se basant sur les implémentations précédentes implémenter les fonctions qui permettent de gérer un arbre équilibrés AVL.

1. Insertion
2. Suppression

Arbre rouges et noirs :

En se basant sur les implémentations précédentes implémenter les fonctions qui permettent de gérer un arbre équilibrés Rouge et Noir.

1. Insertion
2. Suppression

Partie II

Touts les programmes suivants sont à écrire en C++ et en utilisant exclusivement la bibliothèque standard STL.

Les `priority_queue` : les tas

On souhaite gérer une liste de patients qui arrivent aux urgences d'un l'hôpital, l'ordre de passage est déterminé par l'échelle de la douleur qui varie entre $[0, 10]$. Lorsque un patient arrive aux urgences, une infirmière détermine son échelle de douleur est le saisie dans la liste.

1. Créer la classe qui représente un patient.
2. Écrire un programme qui affiche un menu et implémente les fonctionnalités suivantes :
   — Nouveau : ajouter un nouveau patient.
   — Nbr Patient : affiche le nombre de patients restant dans la liste.
   — Passage : affiche le nom du prochain patient à passer chez le médecin et le retire de la liste.

Les `map` : les arbres ordonnés par clé

Écrire un programme qui compte le nombre d'occurrences de chaque mot dans un fichier texte donné en entrée.

1. Le fichier est donné en entrée à l'aide de la ligne de commande `./prog fichier.txt`
2. Le programme demande à l'utilisateur de saisir un mot et affiche en retour le nombre d'occurrences de ce mot dans le fichier texte donné en entrée.

Les `set` : les ensembles

Écrire un programme qui vérifie si un mot existe dans un fichier texte donné en entrée.

1. Le fichier est donné en entrée à l'aide de la ligne de commande `./prog fichier.txt`
2. Le programme demande à l'utilisateur de saisir un mot et affiche en retour si le mot existe dans le texte ou pas.