

Arbres binaires de recherche

Exercice 1

Soit la liste de valeurs $L = \{0, 5, 11, 16, 20, 35, 38, 120, 207, 1018\}$.

1. Construire un arbre binaire de recherche contenant la liste de valeurs L en veillant à minimiser sa hauteur.
2. Proposer un algorithme d'ajout d'un élément dans un ABR, puis calculer sa complexité.
3. Ajouter à l'arbre les valeurs $A = \{1, 10, 15, 618, 1011\}$.
4. Proposer un algorithme de recherche d'un élément dans un ABR, puis calculer sa complexité.
5. Combien de comparaisons avec les éléments de l'arbre doivent être effectuées dans la recherche des valeurs $R = \{3, 16, 58\}$ en partant de la racine de l'arbre à chaque fois.
6. Supprimer de l'arbre les valeurs $S = \{207, 11, 20, 35\}$ en partant du noeud qui contient la valeur à chaque fois.

Exercice 2

On considère le dernier ABR de l'exercice précédent.

1. Proposer des algorithmes récursifs pour les parcours préfixe, infixe et postfixe (ou suffixe), puis calculer leur complexité.
2. Lister les éléments de l'arbre avec des parcours préfixe, infixe et postfixe.
3. Proposer des algorithmes pour énumérer dans l'ordre croissant/décroissant les éléments d'un ABR, puis calculer leur complexité.
4. Proposer des algorithmes pour déterminer les valeurs minimale et maximale d'un ABR, puis calculer leur complexité.
5. Proposer des algorithmes pour déterminer le successeur/prédécesseur d'un élément d'un ABR dans l'ordre croissant, puis calculer leur complexité.

6. Donner le successeur et le prédécesseur des valeurs 1, 5 et 207 et préciser les nœuds de l'arbre par lesquels il faut passer pour trouver ces valeurs.

Exercice 3

On considère le dernier ABR de l'exercice 1.

1. Définir les algorithmes de rotation à droite et à gauche d'un sous-arbre.
2. Évaluer la complexité de ces algorithmes.
3. Appliquer une suite de rotations pour équilibrer l'arbre.

Exercice 4

Soit la liste de valeurs $L = \{0, 1, 5, 10, 15, 16, 38, 120, 618, 1011, 1018\}$.

1. Construire un ABR rouge noir contenant la liste de valeurs L .
2. Proposer un algorithme d'ajout d'un élément dans un ARN, puis calculer sa complexité.
3. Ajouter à l'arbre les valeurs $A = \{60, 150, 200\}$.