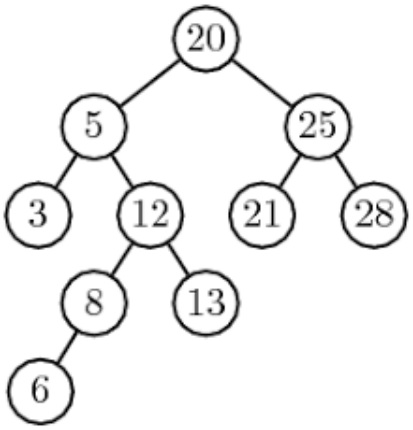
Université Mohammed VI Polytechnique **Année 2021-2022**
Licence LSD **Structures de données**

TRAVAUX DIRIGÉS
SÉRIE 3

1. Arbres binaires de recherche



Exercice 1. *Soit T un arbre binaire de recherche dont les clés sont des entiers. Étant donné un entier x, le successeur de x dans T est défini comme étant le noeud de T ayant la plus petite clé qui est plus grande que x. Par exemple, le successeur de* 10 *dans l'arbre ci-dessus est le noeud ayant 12 comme clé.*

Écrire une fonction `successeur()` *qui prend en entrée x et T et retourne l'adresse du noeud successeur de x dans T. Lorsque x n'a pas de successeur dans T la fonction renvoie NULL.*

Exercice 2 (Code positionnel)**.** *Soit T un arbre binaire de recherche. Le code positionnel d'un entier x qui figure dans T est une liste simplement chaînée L, consituée de* 0 *et de* 1*, qui indique le chemin qu'il faut suivre pour aller de la racine jusqu'au noeud qui contient x. Lorsque x est la clé de la racine alors son code positionnel est la liste vide. Si la liste est non vide, our aller de la racine au noeud qui contient x on suit les règles suivantes.*

- *On se place à la racine de T et on lit la liste du début à la fin.*
- *Si l'élément lu est* 0 *on se déplace à gauche du noeud sur lequel on se trouve.*
- *Si l'élément lu est* 1 *on se déplace à droite du noeud sur lequel on se trouve.*

Date: June 7, 2022.

Par exemple, le code positionnel de 6 *dans l'arbre ci-dessus est la liste* $[0, 1, 0, 0]$. *Lorsque* x *ne figure pas dans l'arbre alors son code positionnel est celui qu'on aurait obtenu si on avait inséré* x *à l'arbre* T.

1. *Écrire une fonction* `codePositionnel()` *qui prend en arguments un entier* x *et un ABR* T *et renvoie le code positionnel de* x *dans* T.
2. *Écrire une fonction* `codeInverse()` *qui prend en argument une liste* L, *constituée de* 0 *et de* 1, *et une arbre* T *et renvoie l'adresse du noeud dont la clé possède* L *comme code positionnel. La fonction renvoie NULL si aucun élément de code posionnel* L *ne figure dans* T.

Exercice 3. *Soit* L *un tableau contenant des entiers qu'on suppose deux à deux distincts. On peut trier par ordre croissant le tableau* L *de la manière suivante.*

- *On gènère un arbre binaire de recherche* T *en inserant les éléments du tableau* L *les uns après les autres.*
- *On gènère ensuite un tableau* M *à partir de l'arbre* T *en utilisant un parcours infixe.*

1. *Écrire une fonction* `abrTableau()` *qui permet de générer un ABR à partir du tableau* L.
2. *Écrire ensuite la fonction* `triAbr()` *qui permet de trier le tableau* L *en utilisant l'arbre généré.*
3. *Quelle est la complexité de cet algorithme de tri ?*