TP 7 : Arbres Cousus

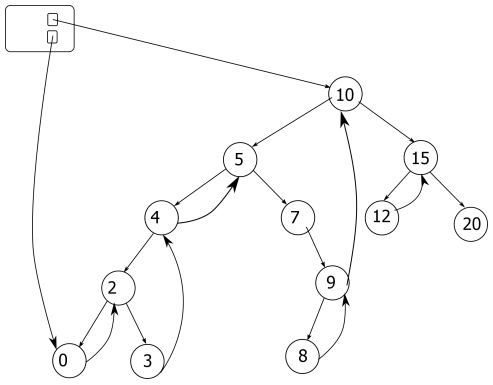
Dans le cadre de l'UE, vous êtes amenés à programmer en **C++** en utilisant les références pour réaliser vos passages de paramètres ainsi que les constructeurs, les destructeurs et les surcharges de l'opérateur d'affectation. Néanmoins, il sera important que vous sachiez toujours basculer dans un autre langage de programmation. Pour cela, il est important que vous établissiez votre raisonnement au niveau du pseudo-langage algorithmique, en utilisant les spécificités du langage utilisé seulement quand vous effectuez la mise en œuvre.

1 Préliminaire

Dans un `Arbre Binaire de Recherche` contenant `n` éléments, quel est le nombre de pointeurs nuls dans les nœuds ?

2 `Arbres Binaires de Recherche Cousus` dans l'ordre infixé

L'idée des arbres cousus consiste à utiliser les pointeurs "non utilisés" d'un `Arbre Binaire de Recherche` pour faciliter le parcours des nœuds dans l'ordre infixé. Ainsi, il sera possible d'utiliser le "pointeur droit" d'un nœud qui n'a pas de sous-arbre droit pour pointer sur le nœud qui lui succède dans l'ordre infixé (voir figure). De la même manière, si l'arbre était doublement cousu, il serait possible d'utiliser le "pointeur gauche" d'un nœud qui n'a pas de sous-arbre gauche pour pointer sur le nœud qui le précède dans l'ordre infixé, mais nous nous contenterons ici de faire des arbres simplement cousus.



2.1 Ajout de données membres dans vos classes

Nous vous proposons d'enrichir votre structure `Arbre Binaire de Recherche` avec une donnée membre permettant de savoir s'il est cousu ou non. Cela peut être fait sous la forme d'un pointeur qui sera nul si l'`Arbre Binaire de Recherche` n'est pas cousu, et qui pointera vers le plus petit nœud de l'arbre si l'arbre est cousu (voir figure). Concernant les nœuds d'un arbre cousu, le problème est de savoir à quel emploi est dédié leur "pointeur droit". Pour cela, vous leur ajouterez une donnée membre permettant de savoir si un nœud a un sous-arbre droit ou pas.

2.2 Etablissement des coutures

Codez la fonction transformant un `Arbre Binaire de Recherche` non cousu en `Arbre Binaire de Recherche` simplement cousu. L'idée consiste à réaliser un parcours en profondeur récursif de l'arbre. Dans le chemin menant au nœud courant n, il est important de conserver l'adresse du nœud le plus bas au dessous duquel on a bifurqué à gauche (on conserve la valeur *nullptr* sinon). C'est l'adresse du nœud qui contient le successeur de n dans l'ordre infixé. Pour chaque nœud n rencontré dont le pointeur droit est nul, on détourne l'utilisation de ce pointeur droit pour le faire pointer vers le nœud contenant son successeur dans l'ordre infixé (ie. nœud dont on a stocké l'adresse à la descente).

Pour cela, vous coderez une fonction interne récursive qui prendra un `Noeud n` en paramètre `Donnée-Résultat`. Ainsi vous pourrez détourner l'utilisation de son "pointeur droit", dans le cas où celui-ci n'est pas utilisé pour pointer vers un sous-arbre. Cette fonction interne récursive prendra également en paramètre l'adresse du `Noeud` le plus bas rencontré sur le chemin menant à `n` et pour lequel le chemin s'est orienté sur la gauche.

2.3 Affichage des éléments dans l'ordre infixé

Vous coderez ensuite une fonction permettant d'afficher, dans l'ordre infixé, les éléments d'un `Arbre Binaire de Recherche` simplement cousu. Ceci peut être réalisé avec un simple pointeur "de travail", sans utiliser de pile ni de procédure récursive : chaque fois que l'on s'oriente vers le fils droit d'un nœud, on descend ensuite le plus loin possible sur sa gauche, pour trouver le prochain élément à afficher. Lorsqu'on a affiché l'élément d'un nœud, on recommence si c'est possible l'opération d'aller un coup vers la droite puis à gauche toute, sinon on utilise le nouveau lien. C'est ainsi que l'on trouve le prochain élément à afficher. Si le pointeur droit est nul, c'est que le parcours est achevé.

2.4 Insertion dans un arbre cousu

Modifiez la procédure d'insertion d'un élément dans un `Arbre Binaire de Recherche` simplement cousu de manière à ce qu'elle mette à jour l'aspect cousu de l'arbre suite à une insertion.