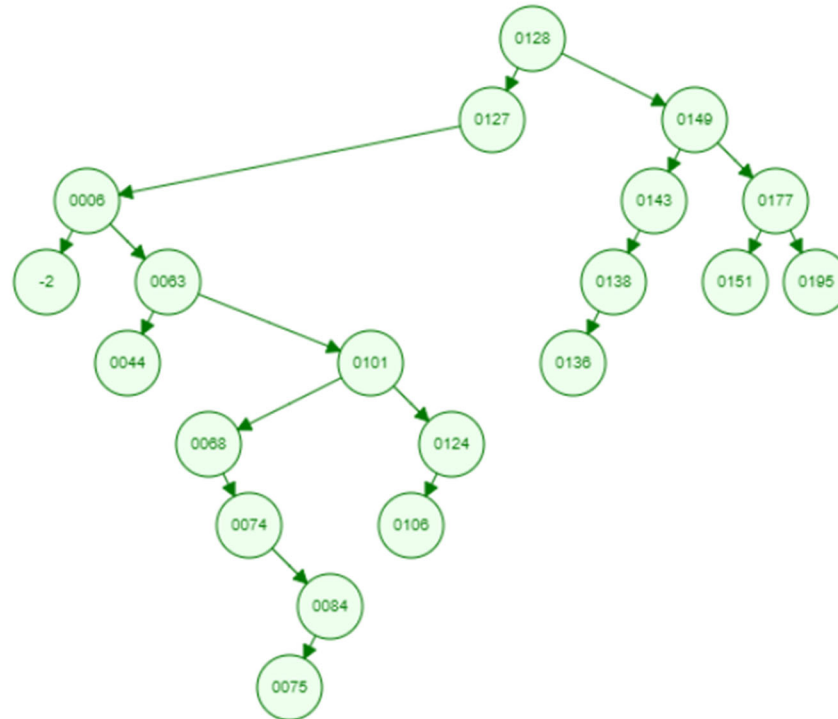


### Corrigé Type – Examen TD – Algo 3

L = {128, 149, 127, 6, 63, 143, 101, 177, 151, 138, 124, -2, 136, 68, 74, 106, 195, 84, 75, 44}

1) L'arbre binaire de recherche (ABR) : (2pts)



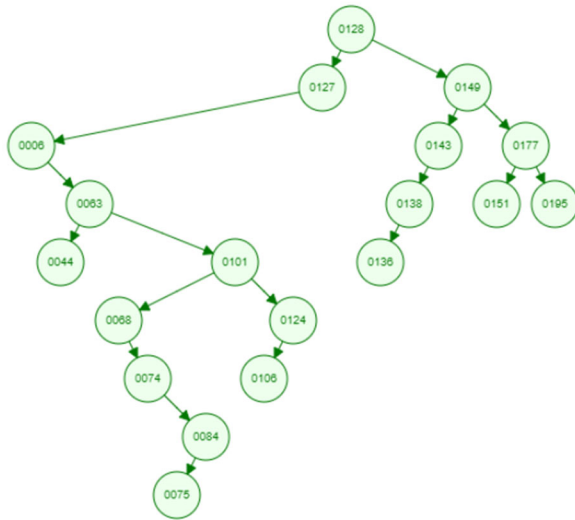
2) Postfixe : On visite d'abord récursivement les sous-arbres gauche et droit, puis la racine. (1 pt)

3) Parcours : (1.5 pts pour chaque type)

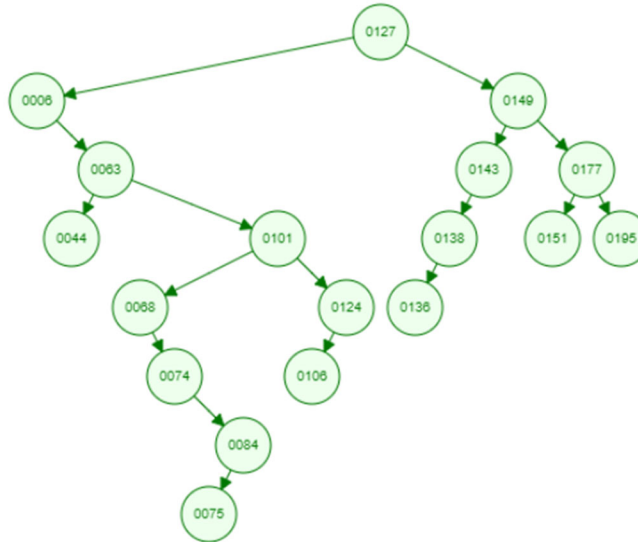
- preorder (prefixe) : 128 127 6 -2 63 44 101 68 74 84 75 124 106 149 143 138 136 177 151 195
- inorder (infixe) : -2 6 44 63 68 74 75 84 101 106 124 127 128 136 138 143 149 151 177 195
- postorder (Postfixe) : -2 44 75 84 74 68 106 124 101 63 6 127 136 138 143 151 195 177 149 128
- breadth-First (Largeur) : 128 127 149 6 143 177 -2 63 138 151 195 44 101 136 68 124 74 106 84 75

#### 4) Suppression :

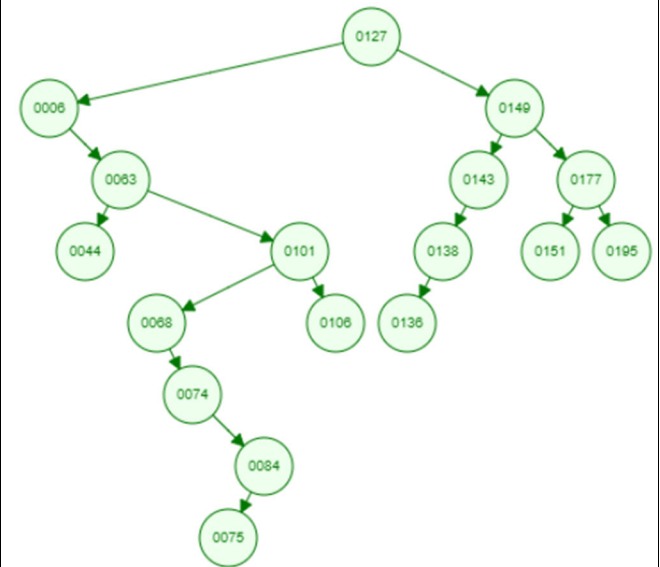
Suppression : -2 (1.5 pt)



Suppression : 128 (1.5 pt)



Suppression : 124 (1.5 pt)



#### 5) La fonction récursive : postfixe (3.5 pt)

```
struct Node { (1.5 pt)
```

```
int value;
```

```
Node *left;
```

```
Node *right;
```

```
};
```

```
void postorderTraversal(Node* node) { (0.2 pts)
```

```
if (node != nullptr) {
```

```
    postorderTraversal(node->left);
```

```
    postorderTraversal(node->right);
```

```
    cout << setw(6) << node->value ;
```

```
}
```

```
}
```

#### 6) L'arbre AVL (0.3 pts)

