



uOttawa

**CSI 2510 - Structures de données et algorithmes**

AUTOMNE 2021 SECTION A

PROFESSEURE DORRA RIAHI

---

## Laboratoire 3

---

Kien Do (ID: 300163370)

1. (a) Puisqu'une propriété d'un arbre est  $n = 2e - 1$  où  $n$  est le nombre de noeuds et  $e$  est le nombre de noeuds externes, on a que  $e = (n+1)/2$ . Et puisqu'il y a 2567 noeuds,  $e = (2567+1)/2 = 1284$ .
- (b) Puisqu'une propriété d'un arbre est  $h \geq \log_2(n + 1) - 1$  où  $h$  est la hauteur et  $n$  est le nombre de noeuds, on a que  $h \geq \log_2(2100 + 1) - 1 \approx 10.03$ . Par conséquent,  $h = 11$ .
- (c) Puisque  $n \leq 2^{n+1} - 1$  où  $n$  est le nombre de noeuds,

$$n \leq 2^{n+1} - 1$$

$$n \leq 2^{(11)+1} - 1$$

$$n \leq 2^{12} - 1$$

$$n \leq 4095$$

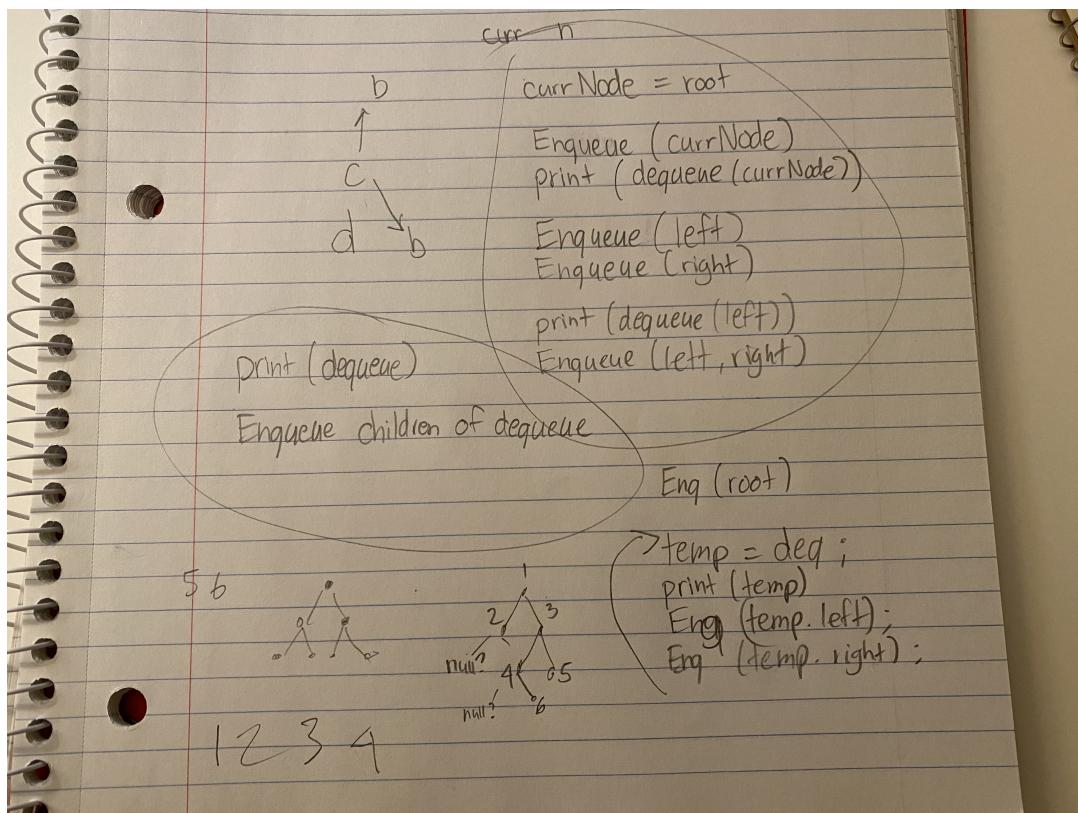
Le nombre maximum de noeuds qu'il peut y avoir dans un arbre binaire complet de hauteur 11 est 4095.

2. (a) i. Pré-ordre: ABDHEJCGO  
ii. Post-ordre: HDJEBOGCA  
iii. In-ordre: HDBJEACGO
- (b) Voici le pseudocode:

```

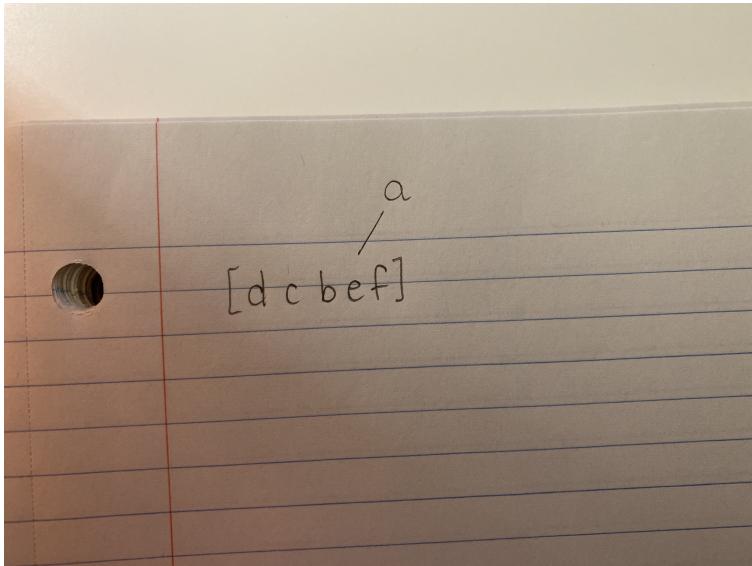
enqueue(root)
while (queue.size > 0):
    currNode = dequeue
    print(currNode.value)
    if (currNode.leftChild is not null)
        enqueue(currNode.leftChild)
    if (currNode.rightChild is not null)
        enqueue(currNode.rightChild)
  
```

Voici mon travail brouillon

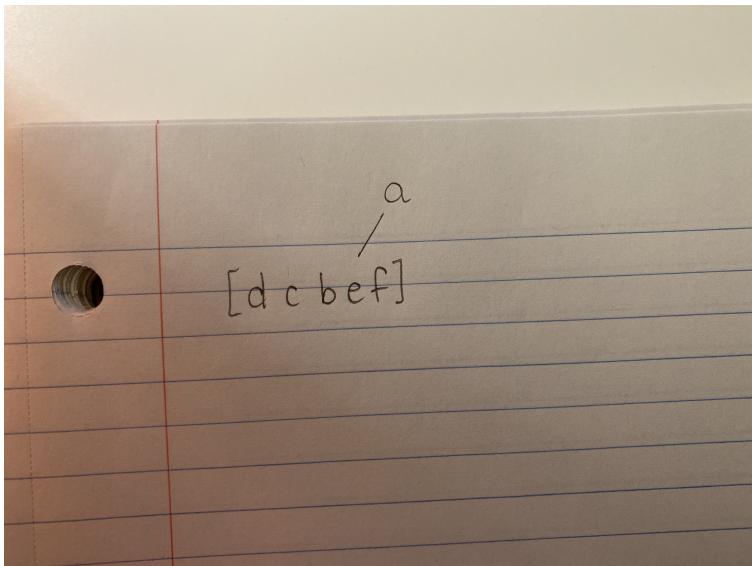


- (c) Inordre = [d c b e f a]  
 Postordre = [d c f e b a]

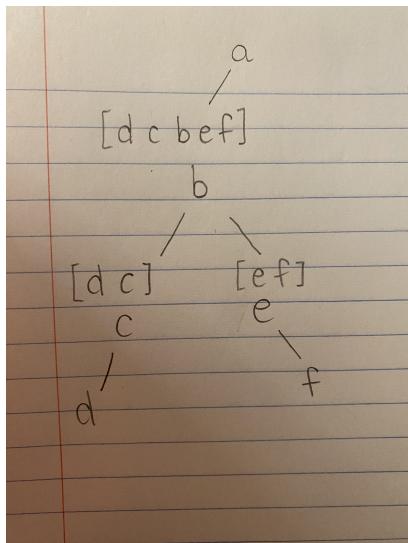
Puisque le parcours postordre met la racine à la fin, on peut déterminer que "a" est la racine. Maintenant qu'on sait la valeur de la racine, on peut déterminer les enfants gauches et droites de cet arbre en voyant la liste inordre. Toutes les enfants sur la gauche de *a* dans la liste inordre viennent à la gauche, et toutes les enfants sur la droite viennent à la droite. Par conséquent, les noeuds enfants gauches sont [d c b e f] et il n'y a pas de noeuds enfants sur la droite parce que le noeud *a* est déjà le noeud le plus à droite dans la liste.



Maintenant, on revoit la liste postordre et on trouve le prochain noeud. Dans ce cas, c'est le noeud *b*. Donc, on sait l'enfant gauche de *a* est *b*. Mais quel sont les enfants gauches et droites de *b*? Il faut voir les noeuds restant (les noeuds qui ne sont pas encore mis sur l'arbre) dans la liste inordre - les noeuds qui sont à la gauche de *b* viennent à la gauche, et les noeuds qui sont à la droite de *b* viennent à la droite. Donc, en voyant la liste inordre, les enfants gauches de *b* sont [d c] et les enfants droites de *b* sont [e f] (comme déjà mentionné, *a* n'est pas inclus car *a* est déjà mis sur l'arbre).



Si on continue ces étapes, on aura quelque chose comme cela ci-dessous,



Finalement, on termine avec un arbre comme cela. Les listes de parcours inordre et postordre fonctionnent avec cet arbre.

