

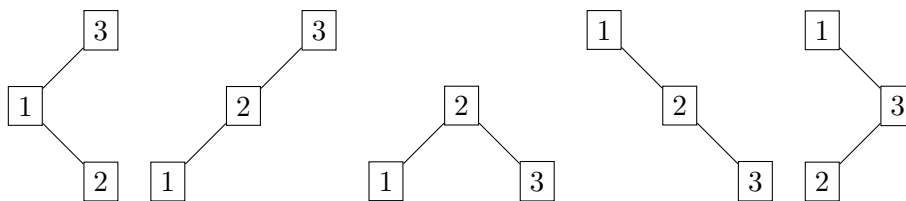
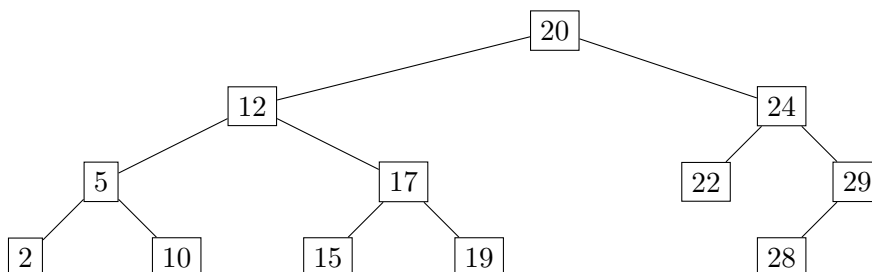
Exercice 1 :**Exercice 2 :**

FIGURE 1 – Un Arbre Binaire de Recherche (ABR)

[2, 5, 10, 12, 15, 17, 19, 20, 22, 24, 28, 29]

Exercice 3 : Retrouver la correction sur le site <https://cviroulaud.github.io>.**Exercice 4 :** Retrouver la correction sur le site <https://cviroulaud.github.io>.

Si l'arbre est équilibré, l'insertion de chaque élément a une complexité proportionnelle à la hauteur de l'arbre soit $O(\log(n))$ dans le pire des cas. Donc la complexité de construction de l'arbre est $O(n \cdot \log(n))$. La recherche de l'élément est en $O(\log(n))$. Donc une complexité totale en

$$O(n \cdot \log(n) + \log(n)) \simeq O(n \cdot \log(n))$$

Si le tableau est déjà trié, l'ABR est un peigne :

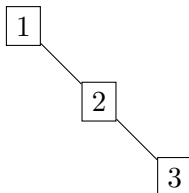


FIGURE 2 – ABR à partir d'un tableau déjà trié

La construction de l'arbre est alors quadratique et la recherche dépend de n .