

**Exercice 1 :** Donner tous les ABR formés de trois nœuds contenant les entiers 1, 2, 3.

**Exercice 2 :**

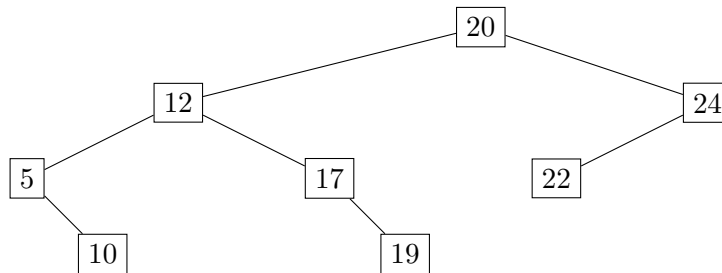


FIGURE 1 – Un Arbre Binaire de Recherche (ABR)

1. Compléter cet ABR en insérant dans l'ordre les valeurs 2, 15, 29, 28.
2. Donner le résultat d'un parcours infixe de cet ABR.

**Exercice 3 :**

1. Écrire la classe **Noeud** et son constructeur. Ce dernier possédera trois attributs :
  - l'entier **valeur** initialisé par un paramètre **v**,
  - le nœud **gauche** initialisé à **None**,
  - le nœud **droit** initialisé à **None**.
2. Créer une instance **arbre** de la classe **Noeud** avec l'argument 13.



FIGURE 2 – Racine de l'arbre binaire de recherche

3. Écrire la méthode **insérer(self, v: int) → None** de la classe **Noeud** qui insère récursivement **v** dans le sous-arbre gauche ou droit.
4. Insérer 10 entiers aléatoires distincts dans **arbre**.
5. Écrire la méthode **rechercher(self, v: int) → bool** de la classe **Noeud** qui vérifie récursivement si **v** est présent dans l'arbre.
6. Tester la méthode dans les deux cas de figures (valeur trouvée ou non).
7. Écrire la méthode itérative **minimum(self) → int** qui renvoie la valeur minimale de l'arbre.
8. Écrire la version récursive de **minimum**.
9. Écrire la méthode récursive **infixe(self, noeud: object, parcours: list) → None** qui effectue un parcours infixe dans **arbre**.