

### Exercice 9.7

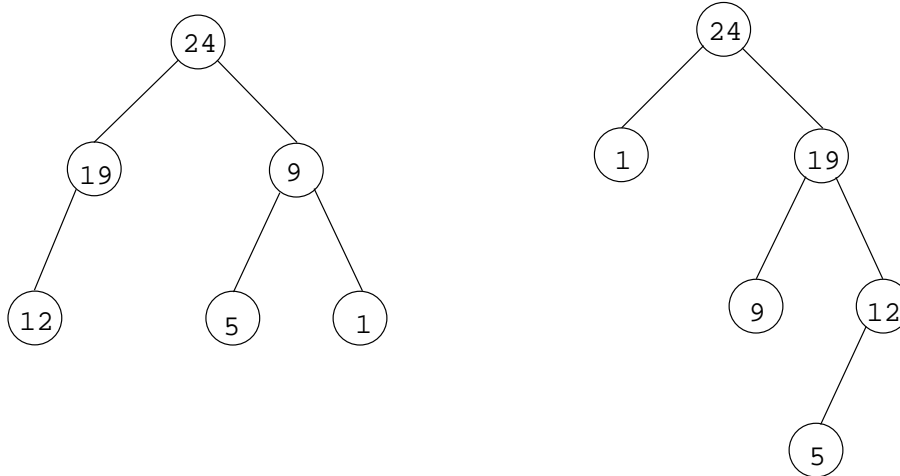
Définir en Java un type d'arbres pour les dictionnaires. Écrire des méthodes de recherche, d'insertion et de suppression pour ces arbres.

## 9.4 Les files de priorité

### 9.4.1 Les arbres partiellement ordonnés

Rappelons qu'une file de priorité est un ensemble dans lequel chaque élément est muni d'une priorité. Les opérations sur un tel ensemble sont la recherche de l'élément de priorité maximale, l'insertion d'un élément et la suppression de l'élément de priorité maximale. Au chapitre 6, nous avons vu que l'on peut représenter une file de priorité par une liste, mais que la recherche et la suppression de l'élément de priorité maximale sont linéaires, à moins que l'on décide d'ordonner les éléments de cette liste par priorité, auquel cas, c'est l'insertion qui devient linéaire.

On peut rendre la recherche, l'insertion et la suppression dans une liste de priorité logarithmiques, en utilisant des arbres, par exemple des arbres de recherche équilibrés. Cependant, il y a plus simple. Un arbre binaire est dit *partiellement ordonné* si le contenu d'un enfant d'un nœud est toujours inférieur au contenu de ce nœud. Par exemple, l'ensemble {1, 5, 9, 12, 19, 24} peut se représenter par les arbres partiellement ordonnés suivants



La recherche de l'élément maximal dans un arbre partiellement ordonné est facile, puisque cet élément est nécessairement le contenu de la racine de l'arbre.