

# Elements d'Algorithmique

## CMTD11 : Tas binaire

---

Mikaël Rabie

Université de Paris, IRIF

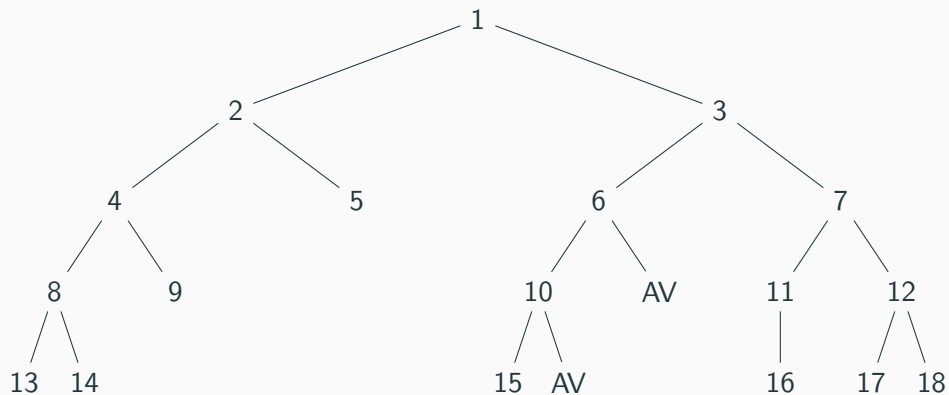


INSTITUT  
DE RECHERCHE  
EN INFORMATIQUE  
FONDAMENTALE



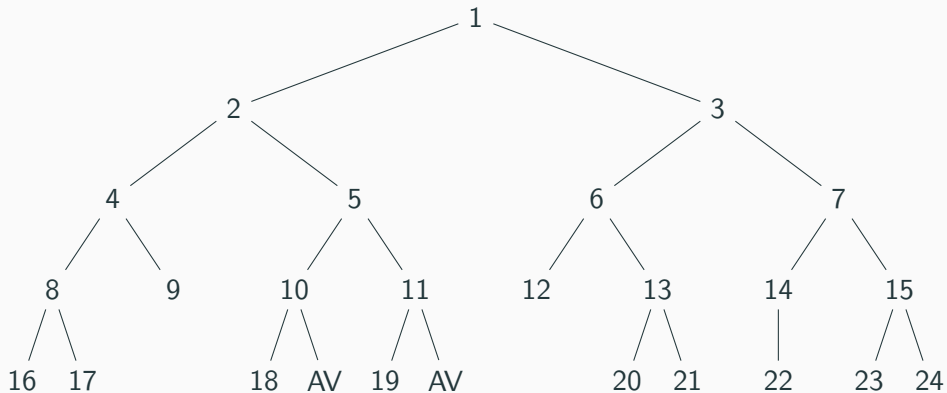
# Arbre Binaire Parfait

- Arbre binaire



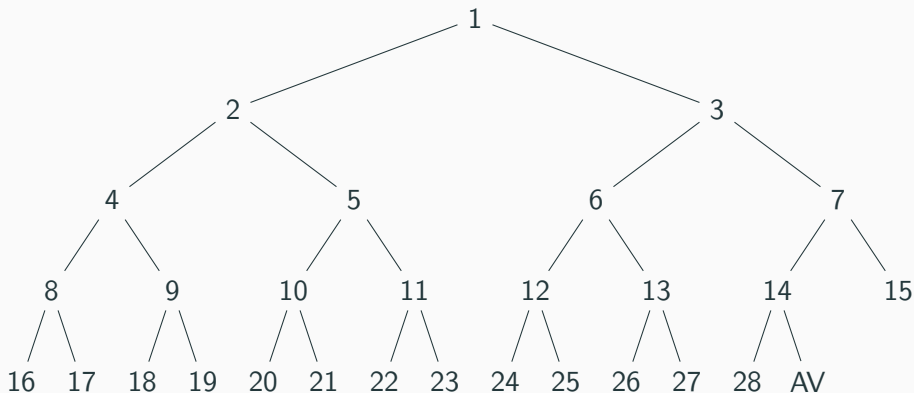
# Arbre Binaire Parfait

- Arbre binaire
- Tous les niveaux sauf le dernier sont remplis



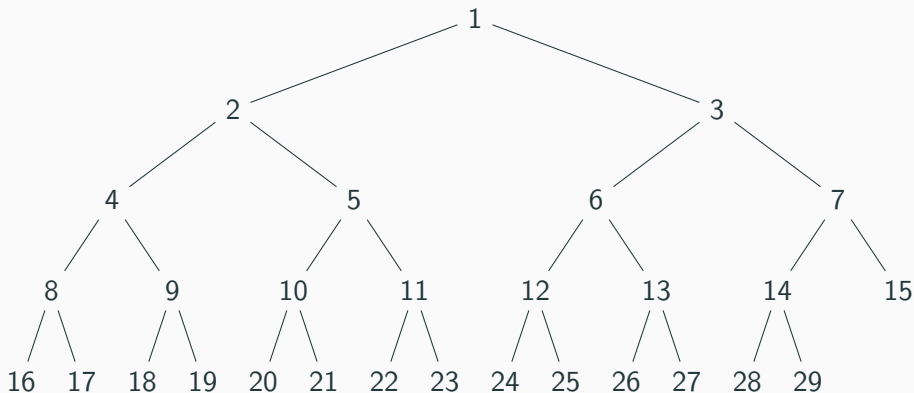
# Arbre Binaire Parfait

- Arbre binaire
- Tous les niveaux sauf le dernier sont remplis
- On remplit un niveau de la gauche vers la droite



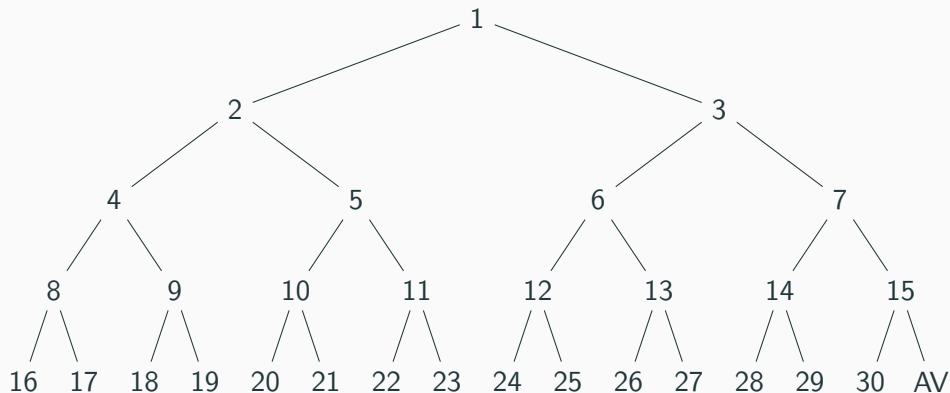
# Arbre Binaire Parfait

- Arbre binaire
- Tous les niveaux sauf le dernier sont remplis
- On remplit un niveau de la gauche vers la droite



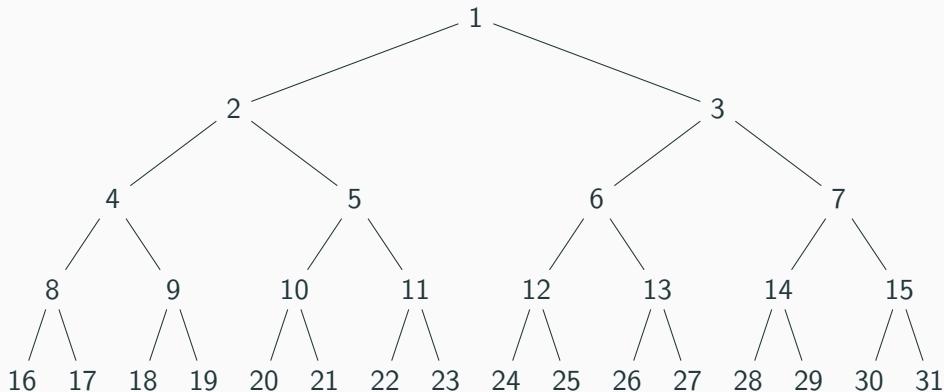
# Arbre Binaire Parfait

- Arbre binaire
- Tous les niveaux sauf le dernier sont remplis
- On remplit un niveau de la gauche vers la droite



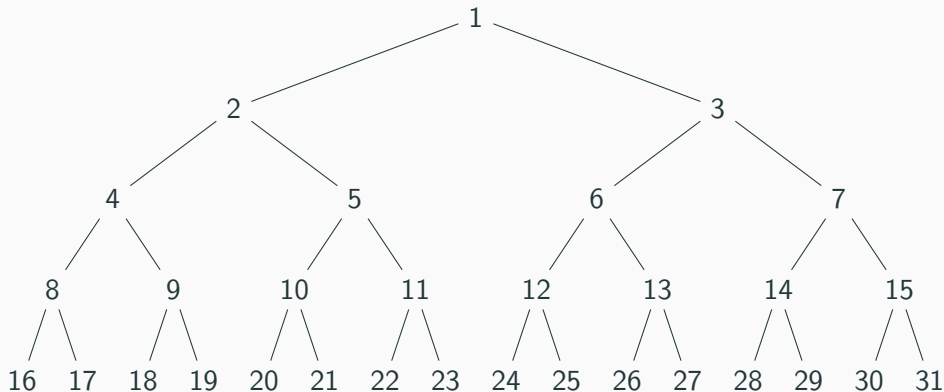
# Arbre Binaire Parfait

- Arbre binaire
- Tous les niveaux sauf le dernier sont remplis
- On remplit un niveau de la gauche vers la droite



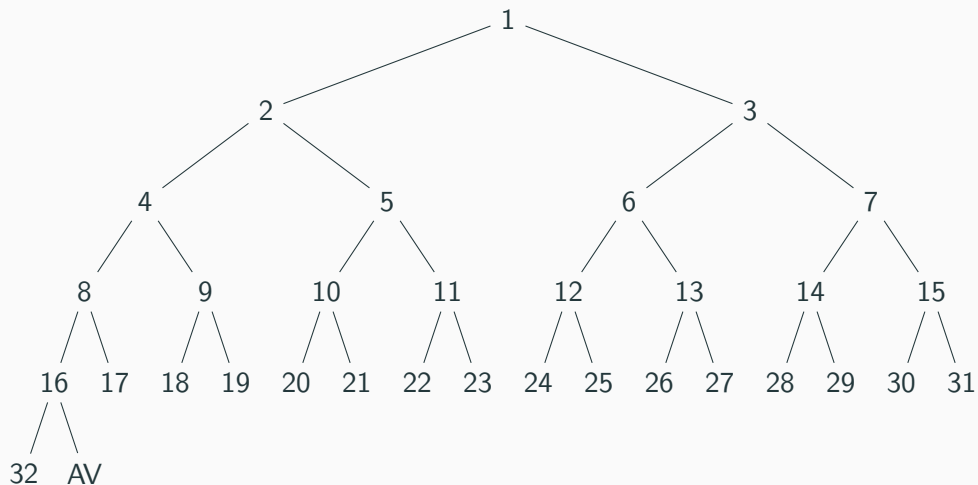
# Arbre Binaire Parfait

- Arbre binaire
- Tous les niveaux sauf le dernier sont remplis
- On remplit un niveau de la gauche vers la droite



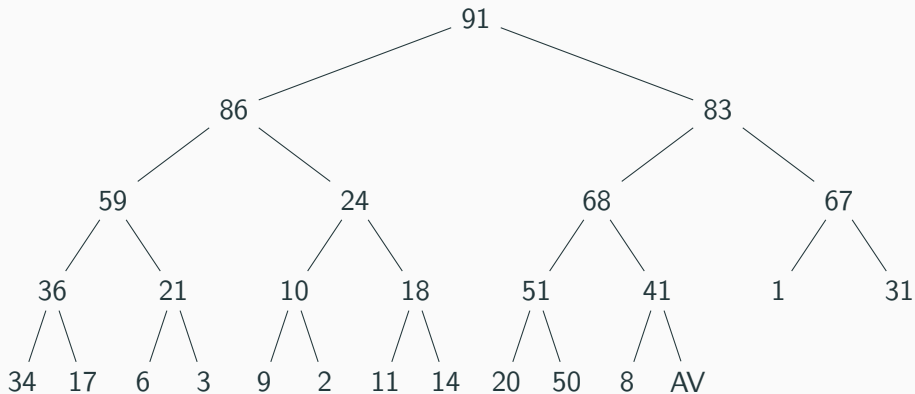


# Arbre Binaire Parfait



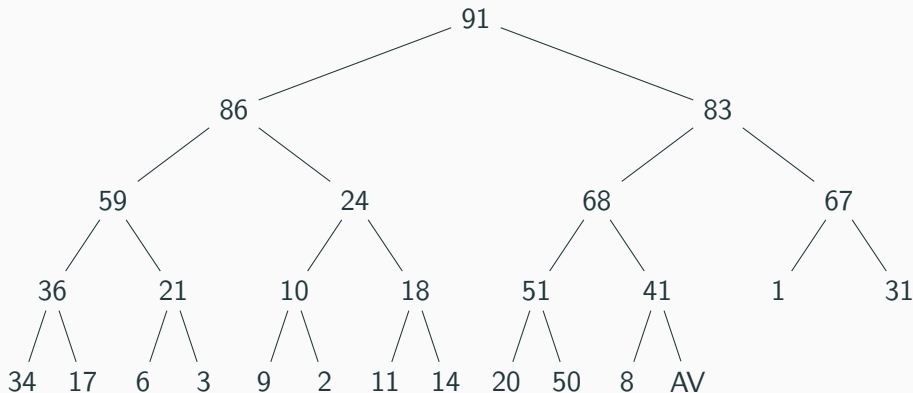
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants



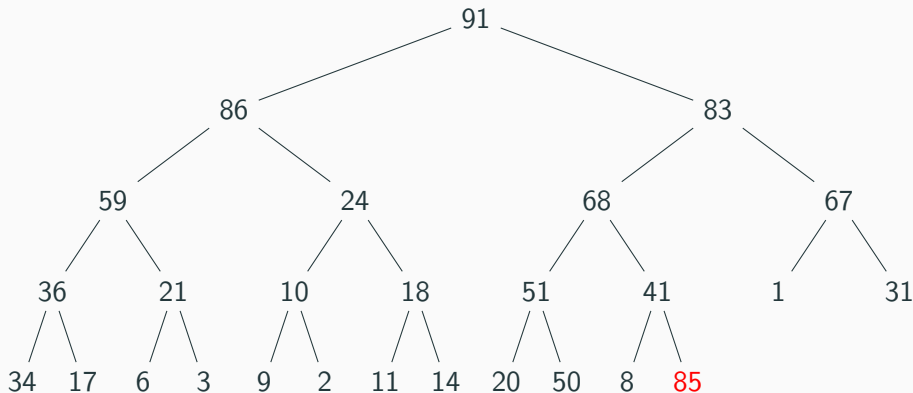
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



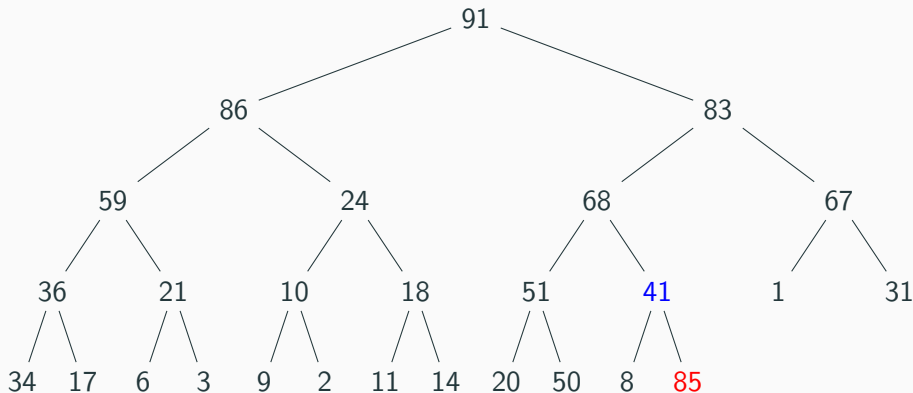
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



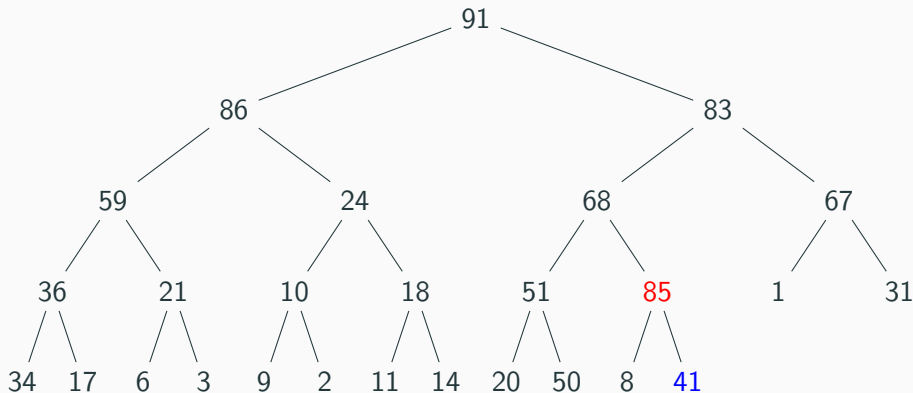
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



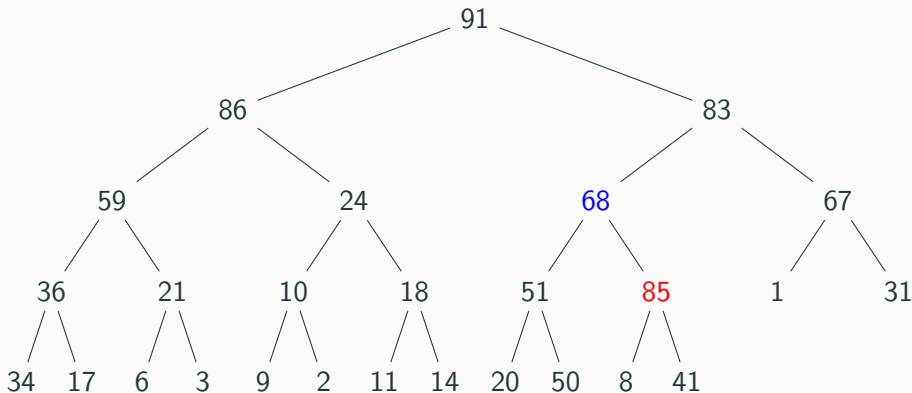
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



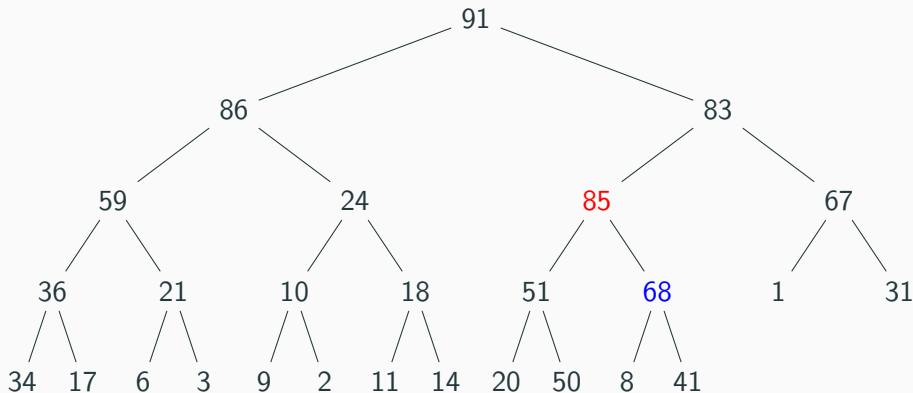
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



# Tas Binaire Max

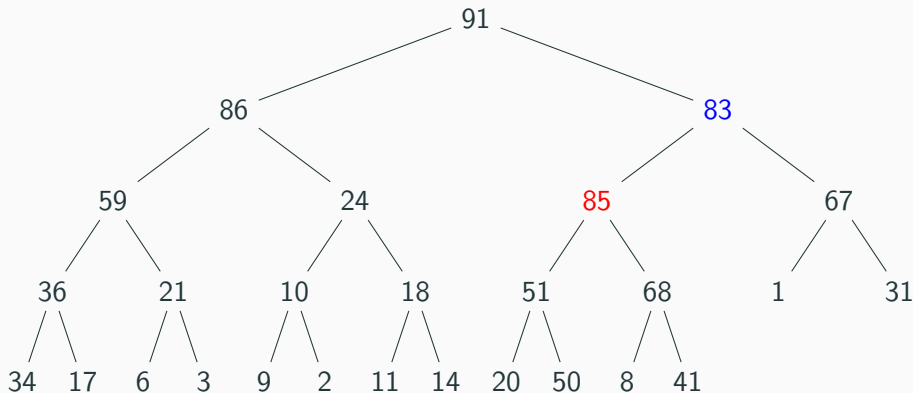
- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax





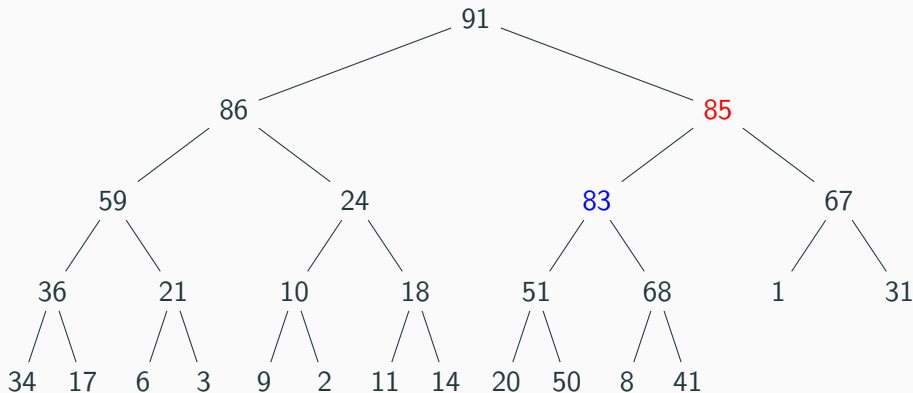
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



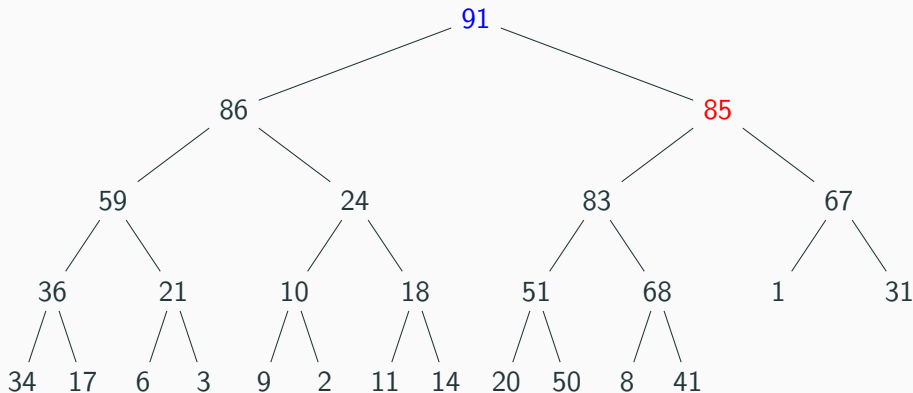
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



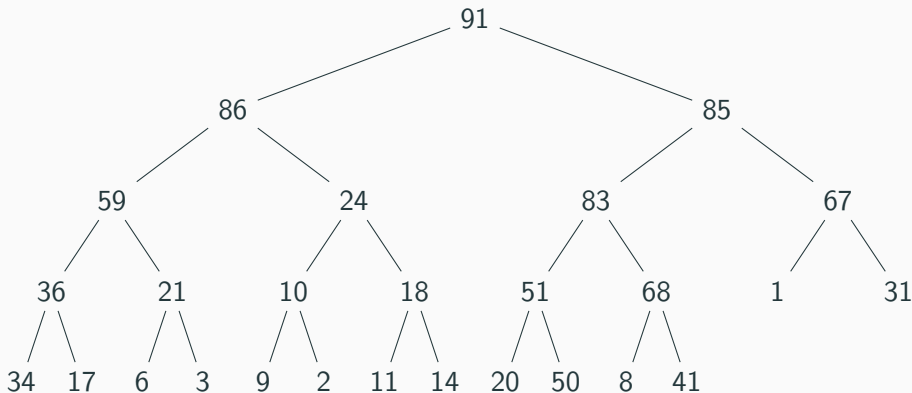
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



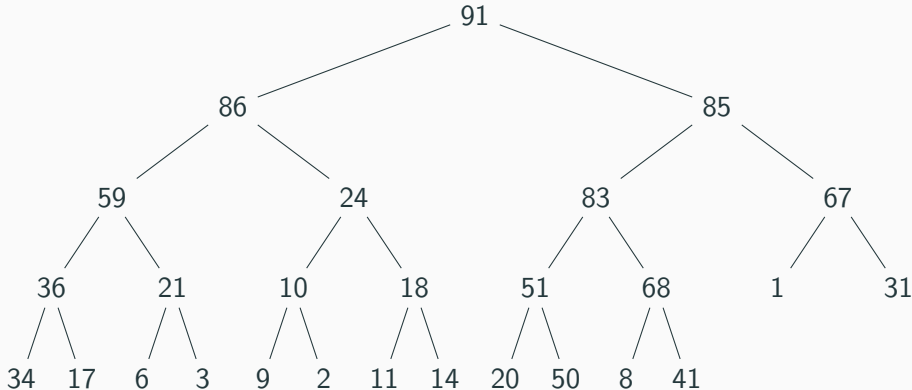
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax



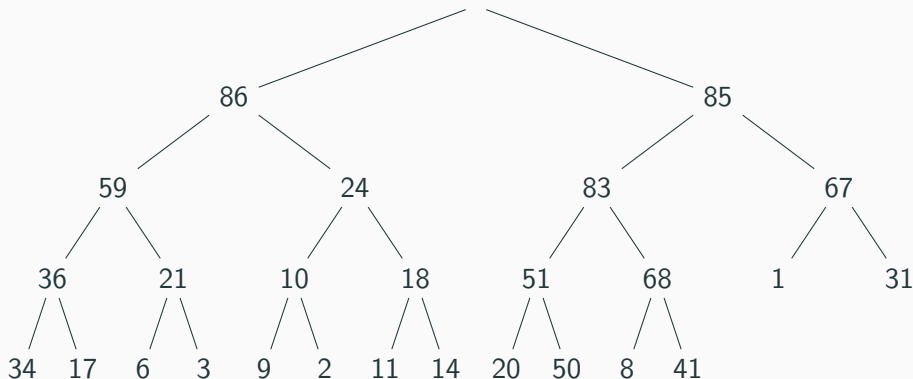
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



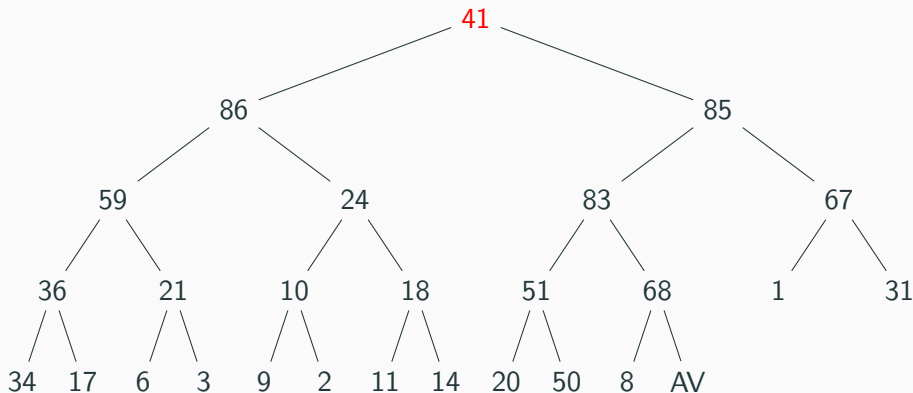
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



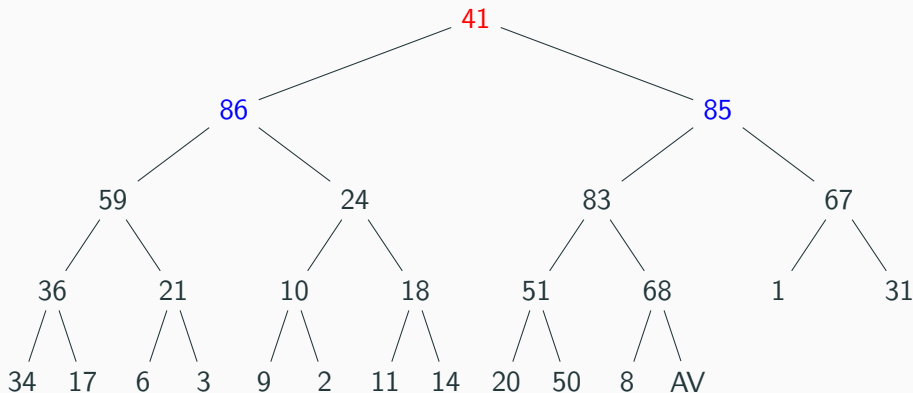
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



# Tas Binaire Max

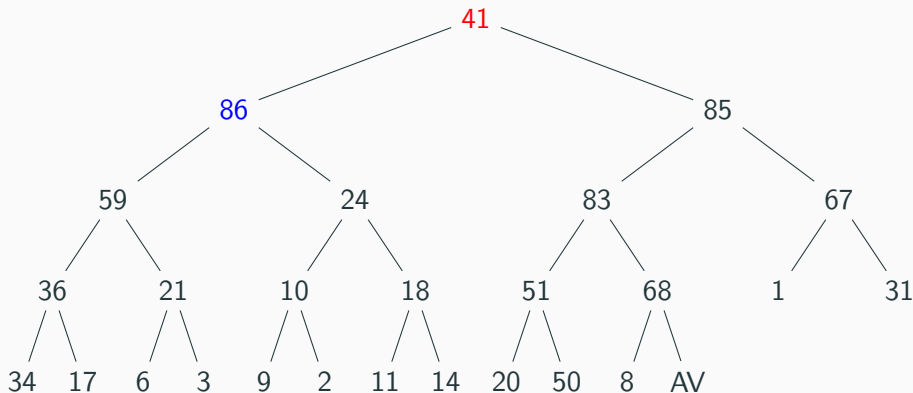
- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)





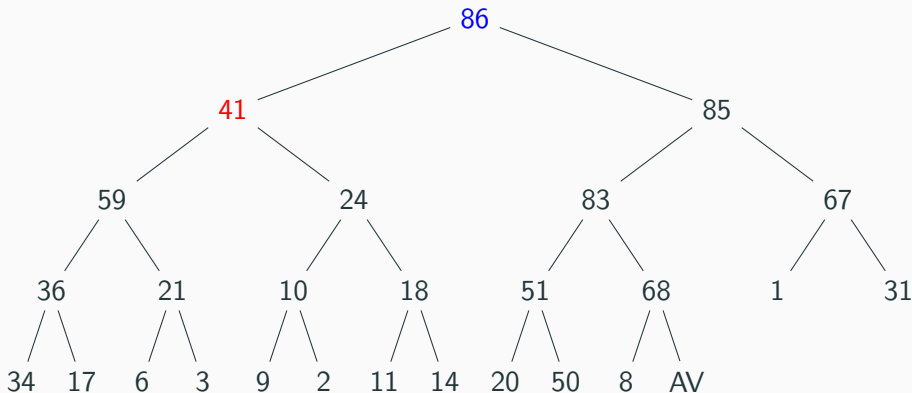
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



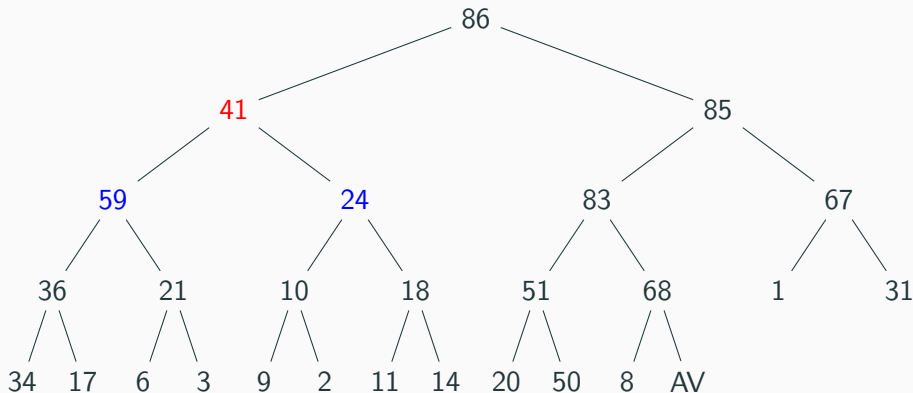
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



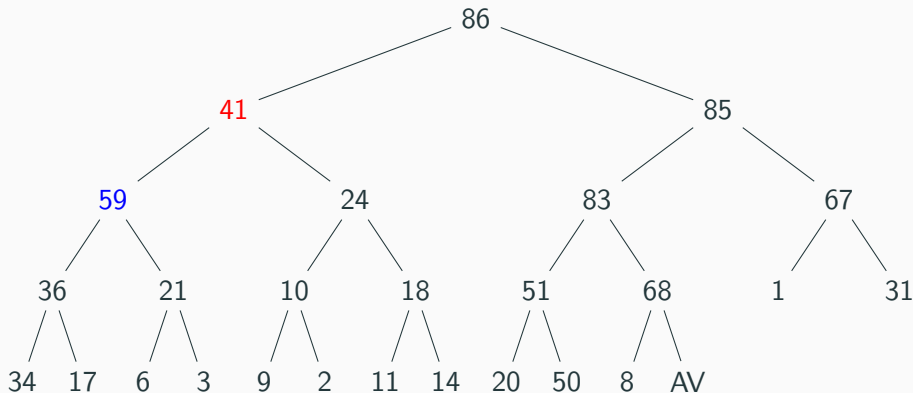
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



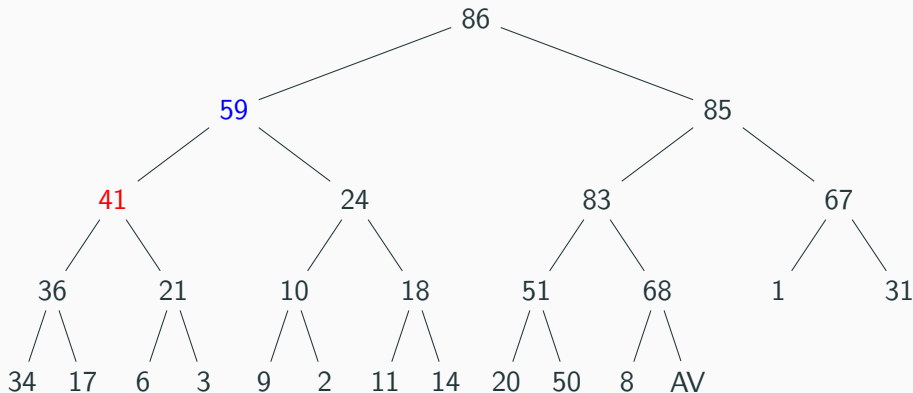
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



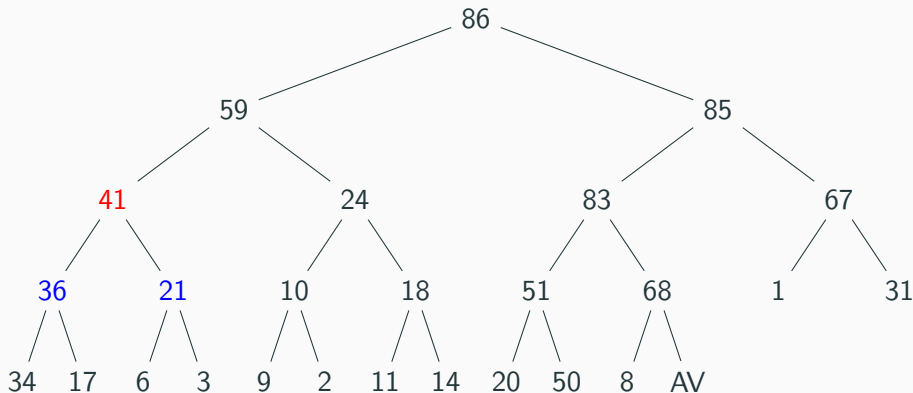
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



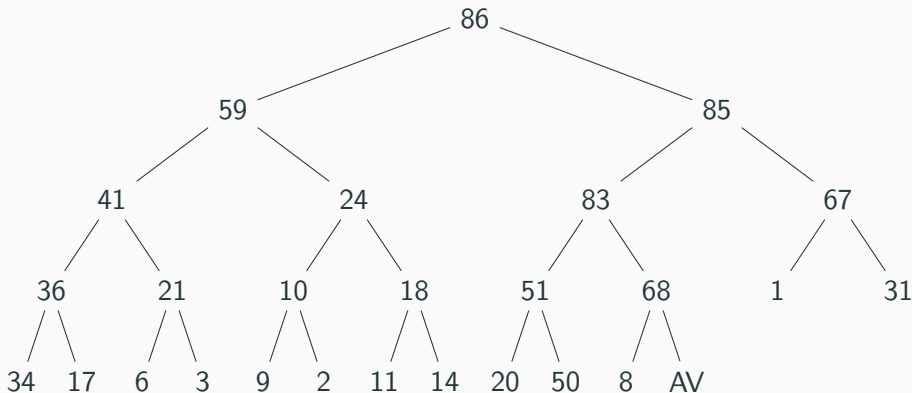
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



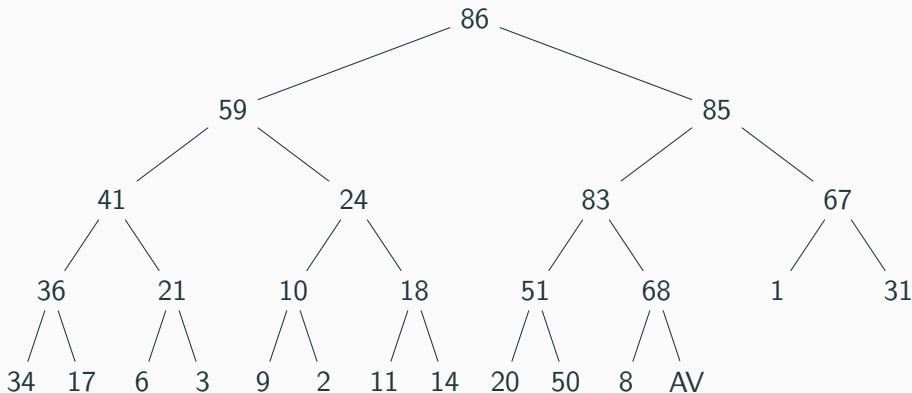
# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, **SuppressionMax** (enlève le max et renvoie sa valeur)



# Tas Binaire Max

- Chaque noeud a un élément plus grand que celui de ses enfants
- Opérations élémentaires : Ajout, SuppressionMax : Complexité  $O(\log n)$





# Tri par Tas

**Entrée :** tableau  $T$  de taille  $n$

```
1: fonction TRITAS( $T$ )  
2:    $t \leftarrow$  TASVIDE()  
3:   pour  $i \leftarrow 0$  à  $n - 1$  faire  
4:     TASAJOUT( $t, T[i]$ )  
5:   pour  $i \leftarrow n - 1$  à  $0$  faire  
6:      $T[i] =$  TASSUPPRESSIONMAX( $t$ )  
7:   retourne  $T$ 
```

# Tri par Tas

**Entrée :** tableau  $T$  de taille  $n$

```
1: fonction TRITAS( $T$ )  
2:    $t \leftarrow \text{TASVIDE}()$   
3:   pour  $i \leftarrow 0$  à  $n - 1$  faire  
4:     TASAJOUT( $t, T[i]$ )  
5:   pour  $i \leftarrow n - 1$  à  $0$  faire  
6:      $T[i] = \text{TASSUPPRESSIONMAX}(t)$   
7:   retourne  $T$ 
```

Complexité :  $2 \cdot (\log 1 + \log 2 + \dots + \log n) = O(n \log n)$