## Arbres binaires de recherche

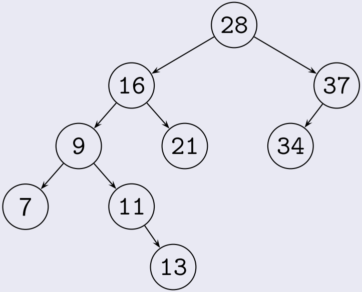
1. Rappeler les relations entre la hauteur $h$ et la taille $n$ d'un arbre binaire.

## Arbres binaires de recherche

1. Rappeler les relations entre la hauteur $h$ et la taille $n$ d'un arbre binaire.
2. Rappeler la définition d'un arbre binaire de recherche (ABR).

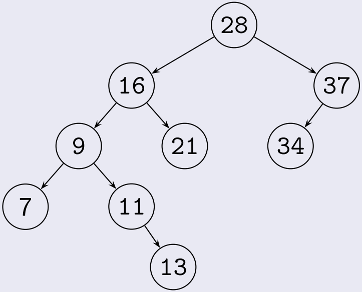
## Arbres binaires de recherche

1. Rappeler les relations entre la hauteur $h$ et la taille $n$ d'un arbre binaire.
2. Rappeler la définition d'un arbre binaire de recherche (ABR).
3. L'arbre ci-dessous est-il un ABR ?



## Arbres binaires de recherche

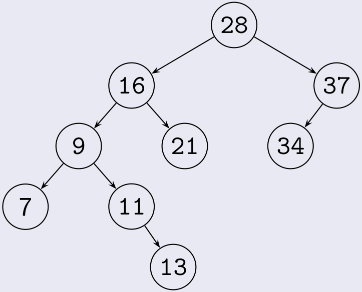
1. Rappeler les relations entre la hauteur $h$ et la taille $n$ d'un arbre binaire.
2. Rappeler la définition d'un arbre binaire de recherche (ABR).
3. L'arbre ci-dessous est-il un ABR ?




4. Quelle est le nombre maximal de comparaison lors de la recherche d'un élément dans cet arbre ?

## Arbres binaires de recherche

1. Rappeler les relations entre la hauteur $h$ et la taille $n$ d'un arbre binaire.
2. Rappeler la définition d'un arbre binaire de recherche (ABR).
3. L'arbre ci-dessous est-il un ABR ?




4. Quelle est le nombre maximal de comparaison lors de la recherche d'un élément dans cet arbre ?
5. Construire un ABR contenant les valeurs $2, 9, 10, 17$ et $21$ et de hauteur minimale. Même question avec la hauteur maximale.

## Complexité

La complexité des opérations d'insertion et de recherche dans un ABR est majorée par la hauteur $h$ de l'arbre.

## Complexité

La complexité des opérations d'insertion et de recherche dans un ABR est majorée par la hauteur $h$ de l'arbre. On descend d'un niveau dans l'arbre à chaque comparaison et la profondeur d'un noeud est inférieure à $h$.

### Complexité

La complexité des opérations d'insertion et de recherche dans un ABR est majorée par la hauteur $h$ de l'arbre. On descend d'un niveau dans l'arbre à chaque comparaison et la profondeur d'un noeud est inférieure à $h$.
Or on sait que $h+1 \leqslant n \leqslant 2^{h+1}-1$, et les deux bornes sont atteintes

## Complexité

La complexité des opérations d'insertion et de recherche dans un ABR est majorée par la hauteur $h$ de l'arbre. On descend d'un niveau dans l'arbre à chaque comparaison et la profondeur d'un noeud est inférieure à $h$.

Or on sait que $h+1 \leqslant n \leqslant 2^{h+1}-1$, et les deux bornes sont atteintes

- Dans le cas d'un peigne $(n = h+1)$ les opérations seront en $\mathcal{O}(n)$.

## Complexité

La complexité des opérations d'insertion et de recherche dans un ABR est majorée par la hauteur $h$ de l'arbre. On descend d'un niveau dans l'arbre à chaque comparaison et la profondeur d'un noeud est inférieure à $h$.
Or on sait que $h+1 \leqslant n \leqslant 2^{h+1}-1$, et les deux bornes sont atteintes

- Dans le cas d'un peigne $(n = h+1)$ les opérations seront en $\mathcal{O}(n)$.
- Dans le cas d'un arbre complet $(n = 2^{h+1}-1)$, les opérations seront en $\mathcal{O}(\log(n))$.

### Complexité

La complexité des opérations d'insertion et de recherche dans un ABR est majorée par la hauteur $h$ de l'arbre. On descend d'un niveau dans l'arbre à chaque comparaison et la profondeur d'un noeud est inférieure à $h$.
Or on sait que $h+1 \leqslant n \leqslant 2^{h+1}-1$, et les deux bornes sont atteintes

- Dans le cas d'un peigne ($n = h+1$) les opérations seront en $\mathcal{O}(n)$.
- Dans le cas d'un arbre complet ($n = 2^{h+1}-1$), les opérations seront en $\mathcal{O}(\log(n))$.

### Définition

Soit $S$, un ensemble d'abres binaires. On dit que les arbres de $S$ sont équilibrés s'il existe une constante $C$ telle que, pour tout arbre $s \in S$ :

$$h(s) \leqslant C \log(n(s))$$

## Rotation d'un ABR

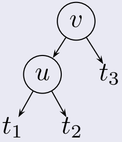
On considère l'ABR suivant où $u$ et $v$ sont les étiquettes des noeuds représentés et $t_1$, $t_2$, $t_3$ des arbres binaires :

# C21 Compléments sur les arbres

## 1. Rappel

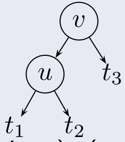
### Rotation d'un ABR

On considère l'ABR suivant où $u$ et $v$ sont les étiquettes des noeuds représentés et $t_1$, $t_2$, $t_3$ des arbres binaires :



## Rotation d'un ABR

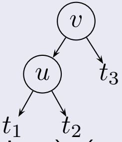
On considère l'ABR suivant où $u$ et $v$ sont les étiquettes des noeuds représentés et $t_1$, $t_2$, $t_3$ des arbres binaires :



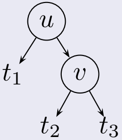

La rotation droite de cet arbre, consiste à réorganiser les noeuds *en conservant la propriété d'*ABR de la façon suivante :

## Rotation d'un ABR

On considère l'ABR suivant où $u$ et $v$ sont les étiquettes des noeuds représentés et $t_1$, $t_2$, $t_3$ des arbres binaires :

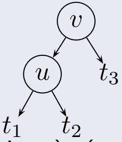



La rotation droite de cet arbre, consiste à réorganiser les noeuds *en conservant la propriété d'*ABR de la façon suivante :

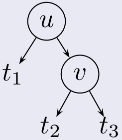


## Rotation d'un ABR

On considère l'ABR suivant où $u$ et $v$ sont les étiquettes des noeuds représentés et $t_1$, $t_2$, $t_3$ des arbres binaires :



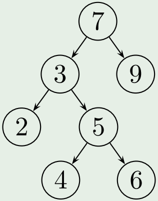

La rotation droite de cet arbre, consiste à réorganiser les noeuds *en conservant la propriété d'*ABR de la façon suivante :




De façon symétrique, la rotation gauche consiste en partant de cet arbre à revenir à l'arbre initial.

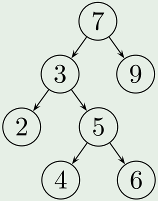
## Exemple

On considère l'arbre binaire suivant :



## Exemple

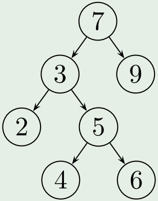
On considère l'arbre binaire suivant :




1. Vérifier qu'il s'agit d'un ABR

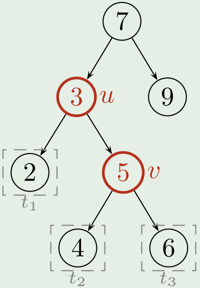
## Exemple

On considère l'arbre binaire suivant :

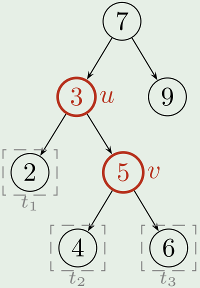



1. Vérifier qu'il s'agit d'un ABR
2. Montrer qu'un utilisant des rotations, on peut transformer cet arbre en un arbre binaire parfait.

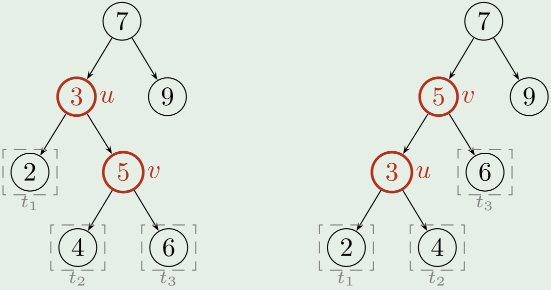
## Correction



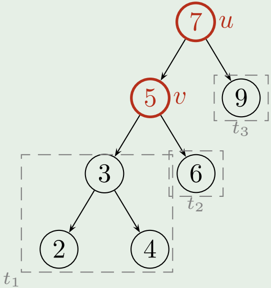
## Correction



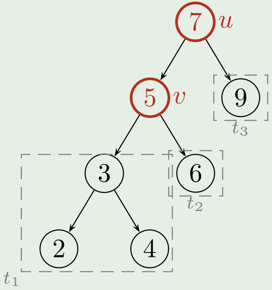
## Correction

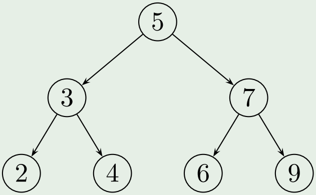


## Correction

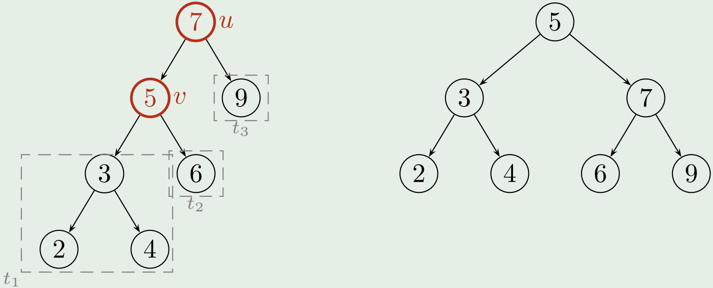


## Correction



## Correction




## Equilibrage d'un abre binaire

Les rotations droite et gauche sont les opérations permettant de maintenir un certain équilibre dans un ABR. Et donc de garantir une complexité logarithmique des opérations usuelles. Parmi les nombreuses possibilités d'ABR équilibrés, nous allons détailler les arbres rouge-noir.

## Définition des arbres rouge-noir

Un arbre rouge-noir $t$ est un ABR (❶), dans lequel chaque noeud porte une information de couleur (rouge ou noir), et ayant les deux propriétés suivantes :

- le père d'un noeud rouge est noir (❷),

## Définition des arbres rouge-noir

Un arbre rouge-noir $t$ est un ABR (➊), dans lequel chaque noeud porte une information de couleur (rouge ou noir), et ayant les deux propriétés suivantes :

- le père d'un noeud rouge est noir (➋),
- le nombre de noeuds noirs le long d'un chemin de la racine à un sous arbre vide est toujours le même (➌), on appellera hauteur noire de $t$ et on notera $b(t)$ cette quantité .

# C21 Compléments sur les arbres

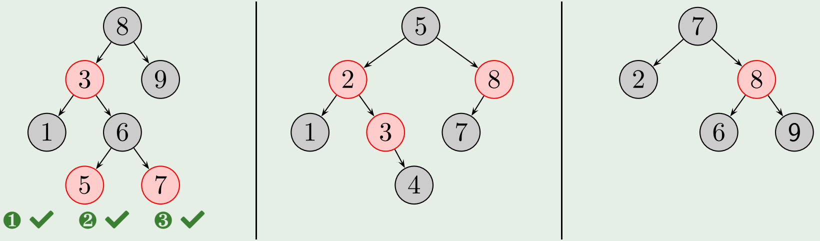
## 2. Arbres rouge-noir

### Définition des arbres rouge-noir

Un arbre rouge-noir $t$ est un ABR (❶), dans lequel chaque noeud porte une information de couleur (rouge ou noir), et ayant les deux propriétés suivantes :

- le père d'un noeud rouge est noir (❷),
- le nombre de noeuds noirs le long d'un chemin de la racine à un sous arbre vide est toujours le même (❸), on appellera hauteur noire de $t$ et on notera $b(t)$ cette quantité .

### Exemples

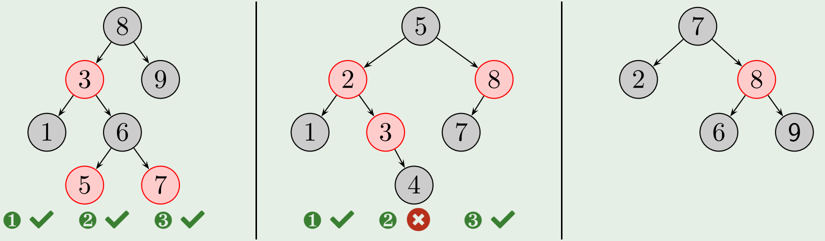


## Définition des arbres rouge-noir

Un arbre rouge-noir $t$ est un ABR (❶), dans lequel chaque noeud porte une information de couleur (rouge ou noir), et ayant les deux propriétés suivantes :

- le père d'un noeud rouge est noir (❷),
- le nombre de noeuds noirs le long d'un chemin de la racine à un sous arbre vide est toujours le même (❸), on appellera hauteur noire de $t$ et on notera $b(t)$ cette quantité .

## Exemples

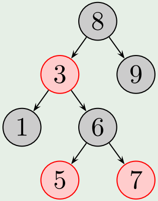


## Définition des arbres rouge-noir

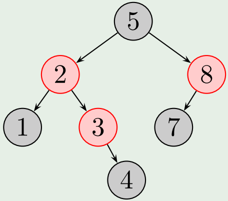
Un arbre rouge-noir $t$ est un ABR (❶), dans lequel chaque noeud porte une information de couleur (rouge ou noir), et ayant les deux propriétés suivantes :

- le père d'un noeud rouge est noir (❷),
- le nombre de noeuds noirs le long d'un chemin de la racine à un sous arbre vide est toujours le même (❸), on appellera hauteur noire de $t$ et on notera $b(t)$ cette quantité .

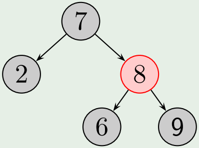
## Exemples




❶ ✓ ❷ ✓ ❸ ✓




❶ ✓ ❷ ✗ ❸ ✓




❶ ✗ ❷ ✗ ❸ ✓

## Proriété d'équilibre des arbres rouge-noirs

Pour tout arbre rouge noir $t$ :

## Proriété d'équilibre des arbres rouge-noirs

Pour tout arbre rouge noir $t$ :

- $h(t) \leqslant 2b(t)$

### Proriété d'équilibre des arbres rouge-noirs

Pour tout arbre rouge noir $t$ :

- $h(t) \leqslant 2b(t)$
- $2^{b(t)} \leqslant n(t) + 1$

## Proriété d'équilibre des arbres rouge-noirs

Pour tout arbre rouge noir $t$ :

- $h(t) \leqslant 2b(t)$
- $2^{b(t)} \leqslant n(t) + 1$

Conséquence : les arbres rouge-noir forment un ensemble d'arbres équilibrés.

## Proriété d'équilibre des arbres rouge-noirs

Pour tout arbre rouge noir $t$ :

- $h(t) \leqslant 2b(t)$
- $2^{b(t)} \leqslant n(t) + 1$

Conséquence : les arbres rouge-noir forment un ensemble d'arbres équilibrés.

## Implémentation

Une implémentation en OCaml sera vue en TP, les opérations d'insertion et de suppression sont difficiles et reposent sur les rotations droite et gauche des ABR.