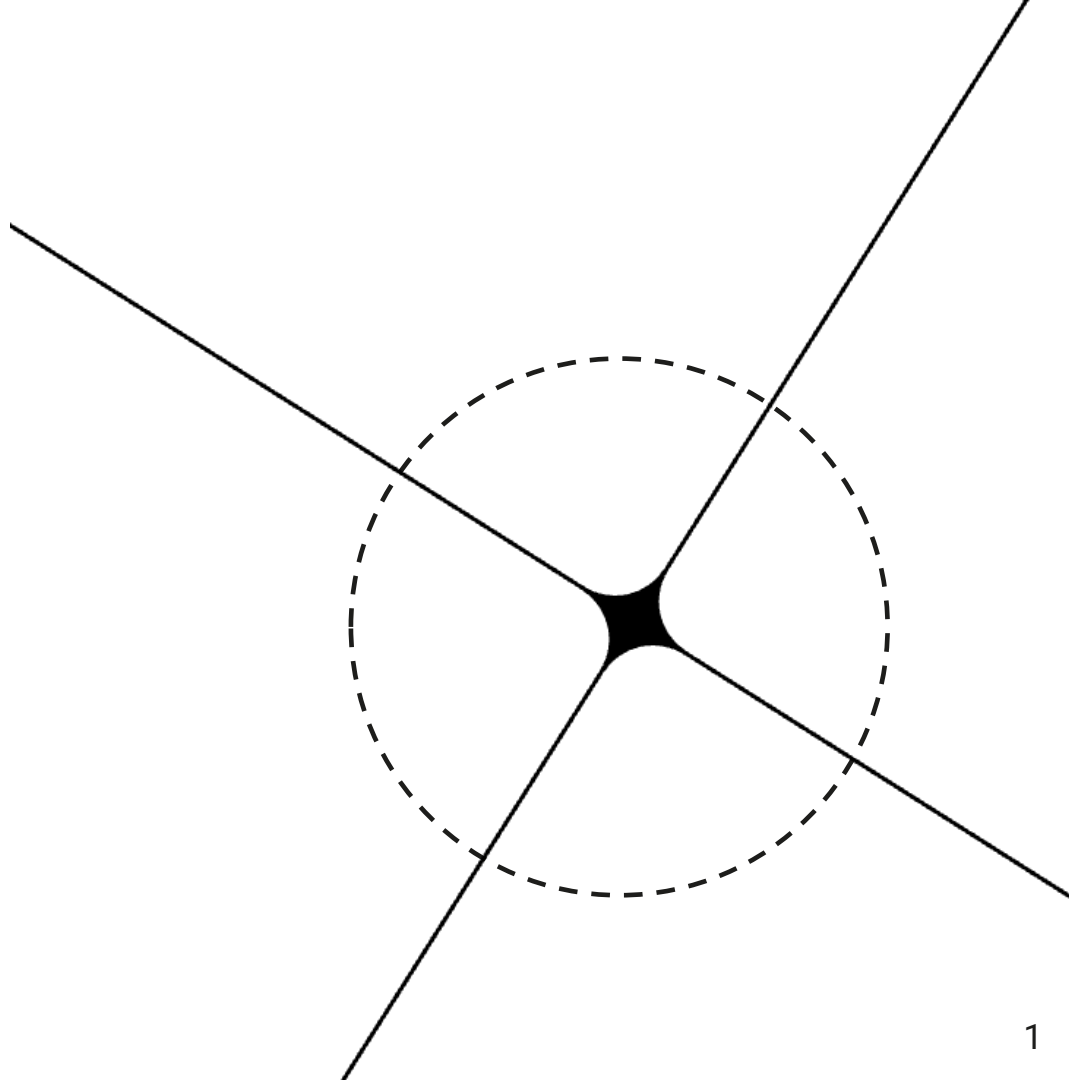


Introduction TP 3

Julien Romero



Rechercher un éléments dans un tableau

- Quelle est la complexité pour rechercher un élément dans un tableau ou une liste **non triés** ?

Rechercher un éléments dans un tableau

- Quelle est la complexité pour rechercher un élément dans un tableau ou une liste **non triés** ?
 - Linéaire : il faut parcourir tous les éléments jusqu'à trouver le bon
- Quelle est la complexité pour rechercher un élément dans un tableau **trié** ?

Rechercher un éléments dans un tableau

- Quelle est la complexité pour rechercher un élément dans un tableau ou une liste **non triés** ?
 - Linéaire : il faut parcourir tous les éléments jusqu'à trouver le bon
- Quelle est la complexité pour rechercher un élément dans un tableau **trié** ?
 - Logarithmique : on peut faire une recherche par dichotomie

Rechercher un éléments dans un tableau

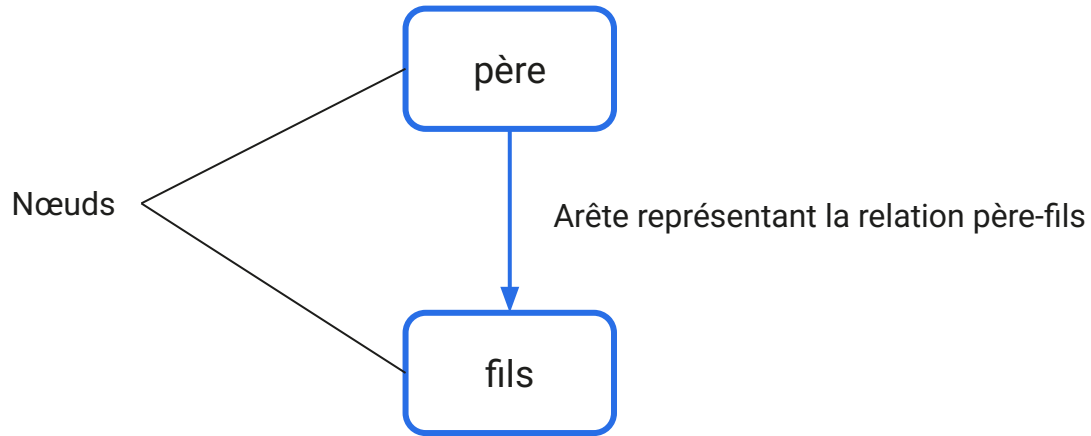
- Quelle est la complexité pour rechercher un élément dans un tableau ou une liste **non triés** ?
 - Linéaire : il faut parcourir tous les éléments jusqu'à trouver le bon
- Quelle est la complexité pour rechercher un élément dans un tableau **trié** ?
 - Logarithmique : on peut faire une recherche par dichotomie

Comment créer une structure de donnée capable de rendre la recherche rapide ?

- On pourrait conserver un tableau trié à chaque ajout d'élément
 - Mais dans ce cas, l'ajout devient coûteux
- On peut utiliser un **arbre binaire de recherche** (entre autres, voir par exemple la skip list)

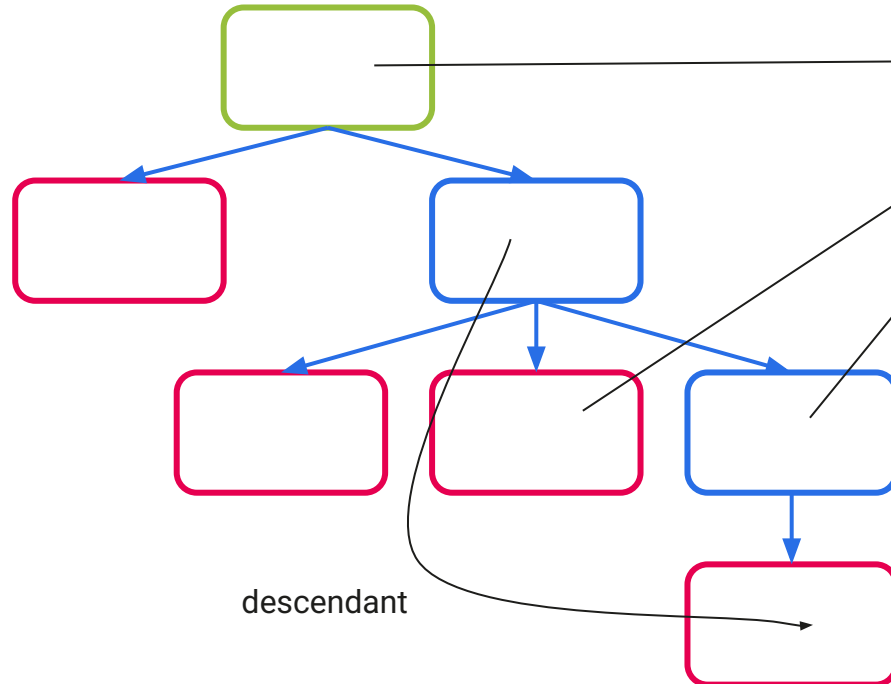
Les arbres en informatique - Nœuds

- Un **arbre** est une structure hiérarchique constituée de **nœuds** connectés avec des arêtes orientées représentant une relation **père-fils**



Les arbres en informatique - Vocabulaire

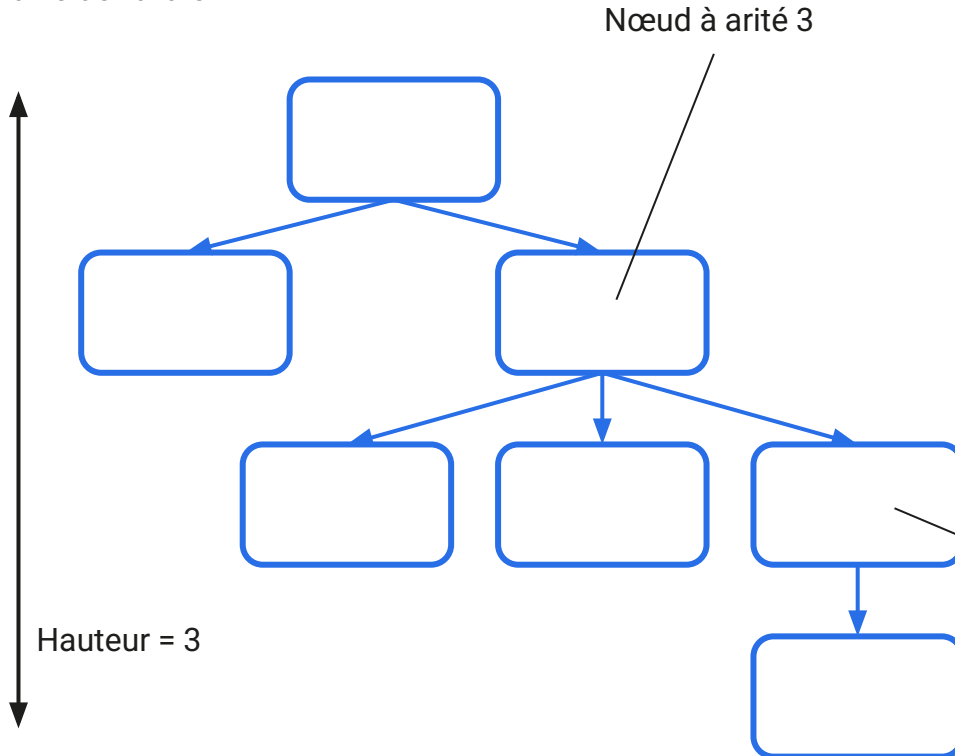
- Dans un arbre, tous les nœuds sont **connectés** (on peut aller d'un nœud à un autre en suivant les arêtes)
- Chaque nœud a au maximum un père (ou parent) et un nombre indéterminé de fils (ou enfant).



- Un nœud sans parent est appelé la **racine de l'arbre** (il y en a exactement une)
- Un nœud sans fils est appelé une **feuille**
- Un nœud qui n'est ni la racine, ni une feuille est appelé un **nœud intermédiaire**
- Les **descendants** d'un nœud sont ses fils et les descendants de ses fils

Les arbres en informatique - Mesures

Taille de l'arbre = 7

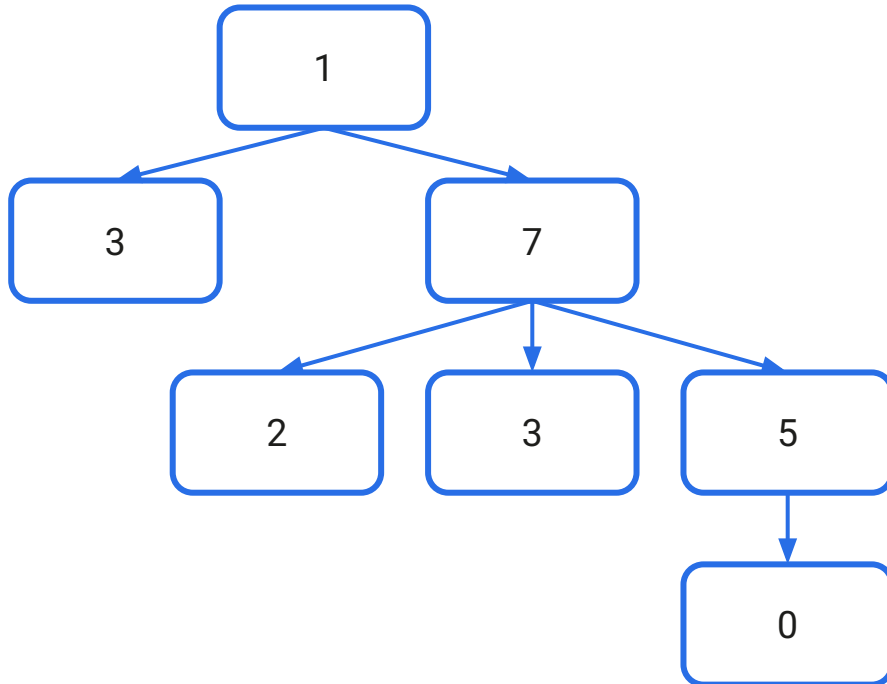


- La **profondeur d'un nœud** est sa distance par rapport à la racine
- La **hauteur** d'un arbre est la profondeur maximale d'un nœud
- La **taille** d'un arbre est son nombre de nœuds
- L'**arité** d'un nœud est son nombre de fils

Nœud à une
profondeur 2

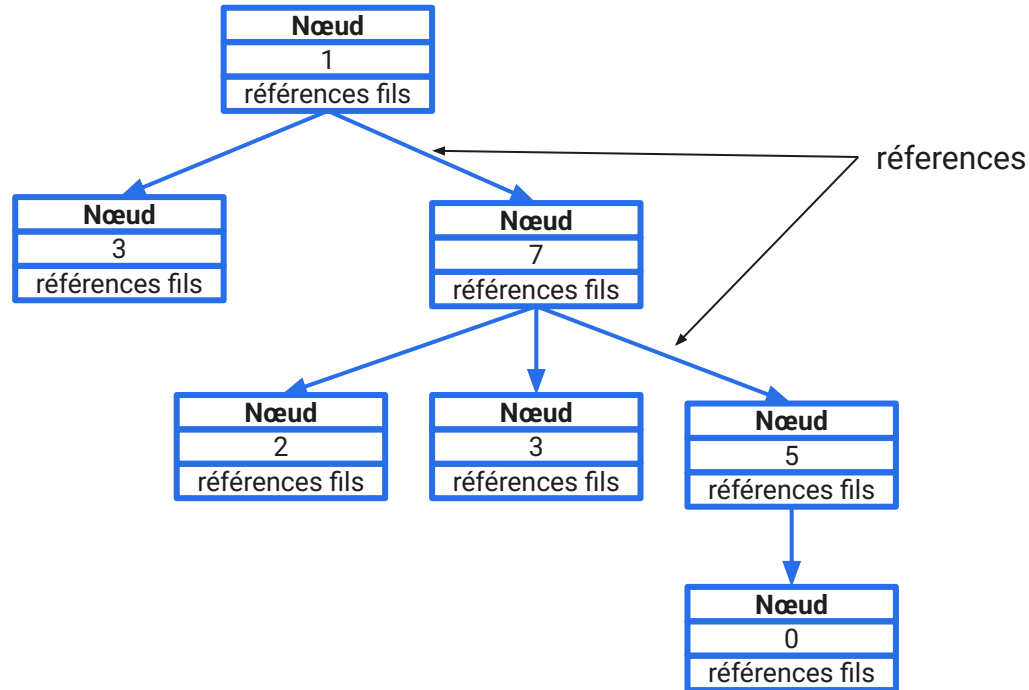
Les arbres en informatique - Étiquette

- On associe souvent une **étiquette** à chaque nœud de l'arbre, c'est-à-dire une valeur.



Les arbres en informatique - Implémentation possible

- L'implémentation va ressembler à celle d'une liste, mais chaque nœud pointera vers 0 ou plusieurs successeurs au lieu d'un seul (ou 0).

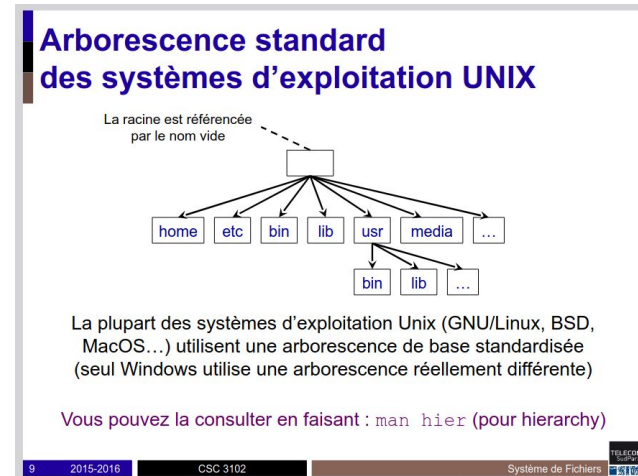
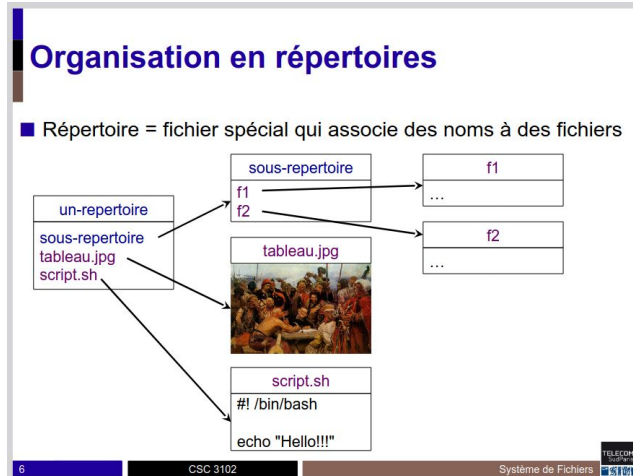


Les arbres en CSC3102 (Bash)

- Où avez-vous vu des arbres en CSC3102 ?

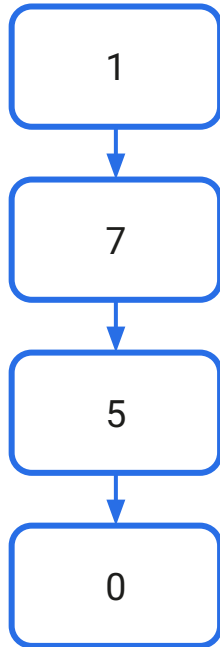
Les arbres en CSC3102 (Bash)

- Où avez-vous vu des arbres en CSC3102 ?
 - Dans l'organisation des fichiers



Cas particulier - Arbre filiforme

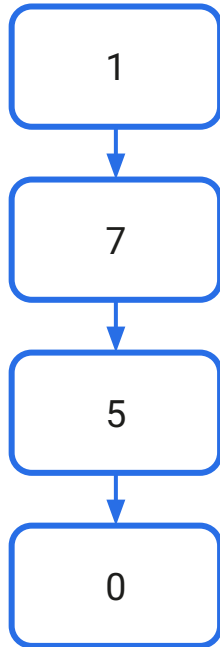
- Un arbre est dit **filiforme** si chaque nœud possède au maximum un fils.



Quelle est la profondeur de l'arbre en fonction du nombre de fils dans ce cas ?

Cas particulier - Arbre filiforme

- Un arbre est dit **filiforme** si chaque nœud possède au maximum un fils.

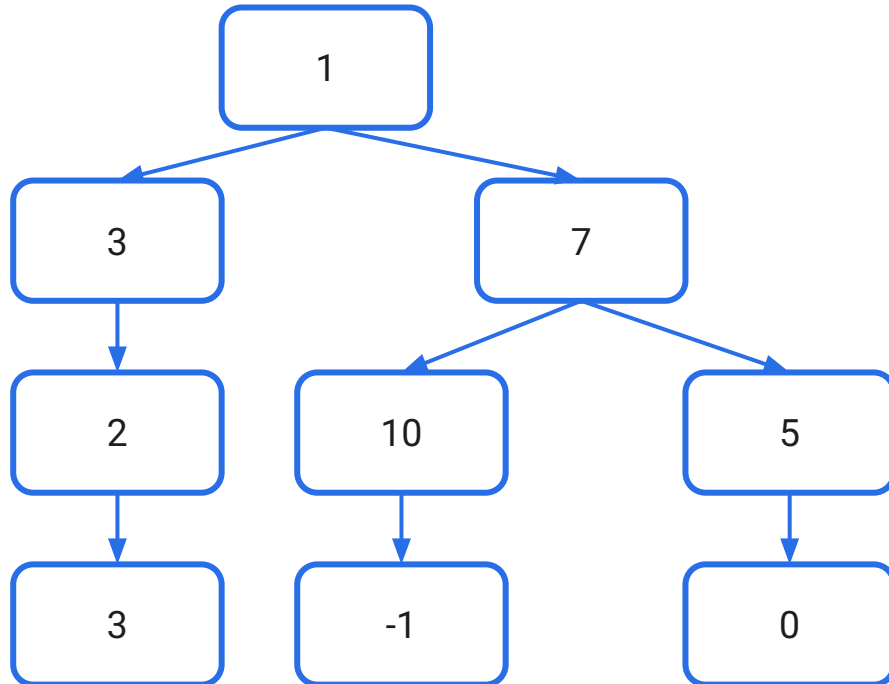


Quelle est la profondeur de l'arbre en fonction du nombre de fils dans ce cas ?

Elle est égale au nombre de fils (-1)

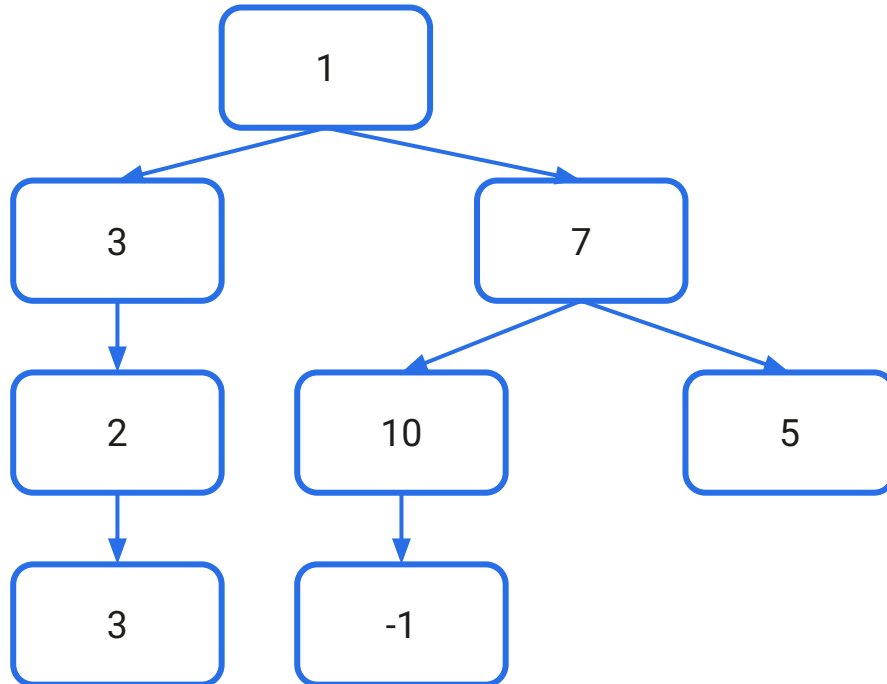
Cas particulier - Arbre équilibré

- Un arbre est dit **équilibré** si toutes ses feuilles sont à la même profondeur



Cas particulier - Arbre équilibré

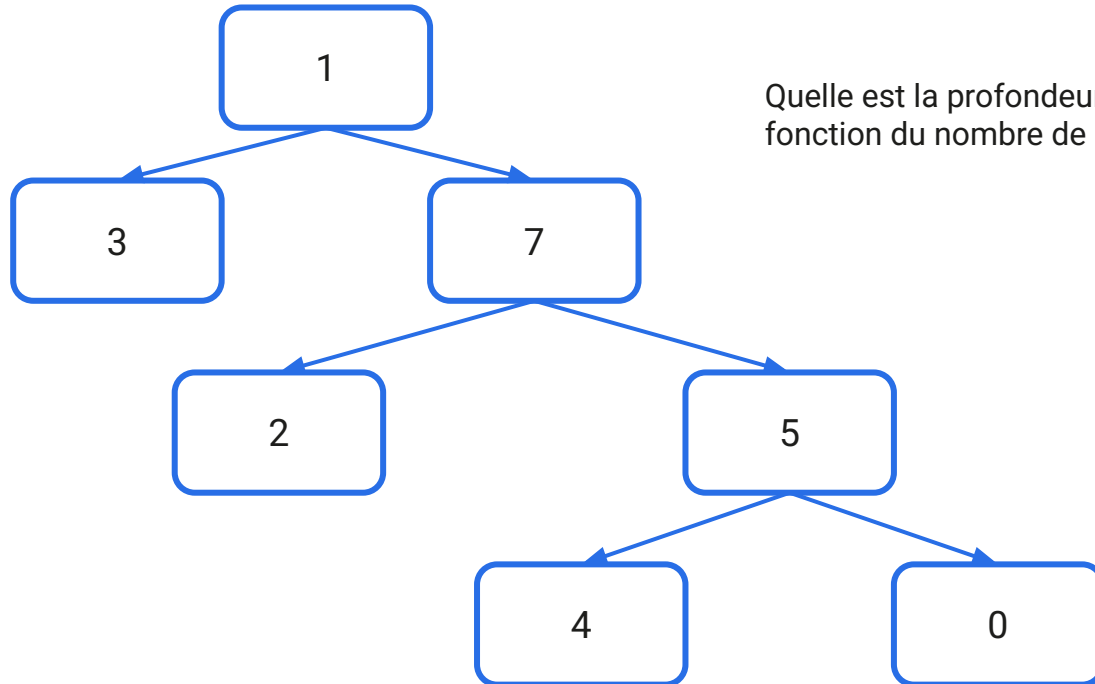
- Un arbre est dit **équilibré** si toutes ses feuilles sont à la même profondeur



Note: Parfois, on permet une différence de 1 par rapport à la profondeur maximale, c'est-à-dire un dernier étage incomplet (c.f. le TP)

Cas particulier - Un arbre binaire (c.f. TP)

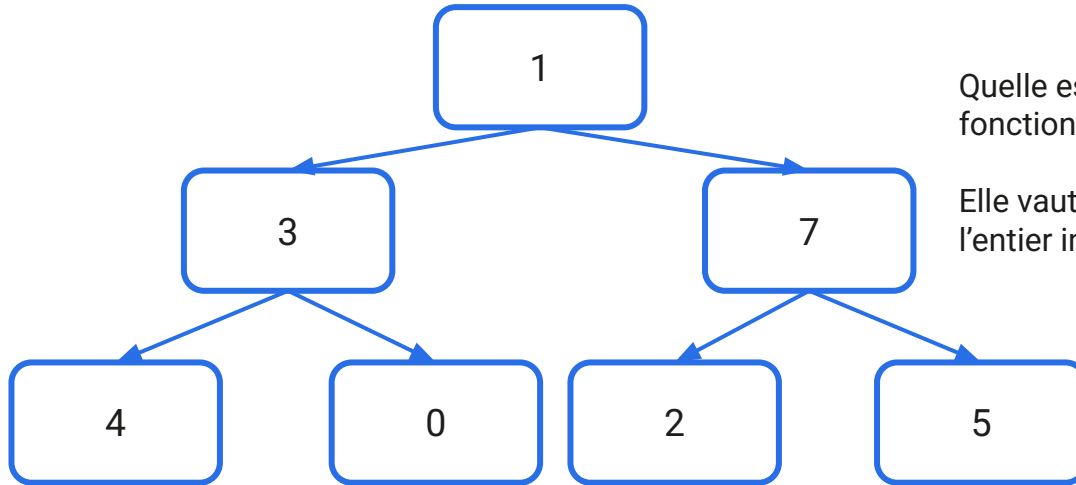
- Un **arbre binaire** est un arbre où chaque nœud possède au maximum 2 fils, appelés fils droit et fils gauche



Quelle est la profondeur d'un arbre binaire équilibré en fonction du nombre de nœuds ?

Cas particulier - Un arbre binaire (c.f. TP)

- Un **arbre binaire** est un arbre où chaque nœud possède au maximum 2 fils, appelés fils droit et fils gauche

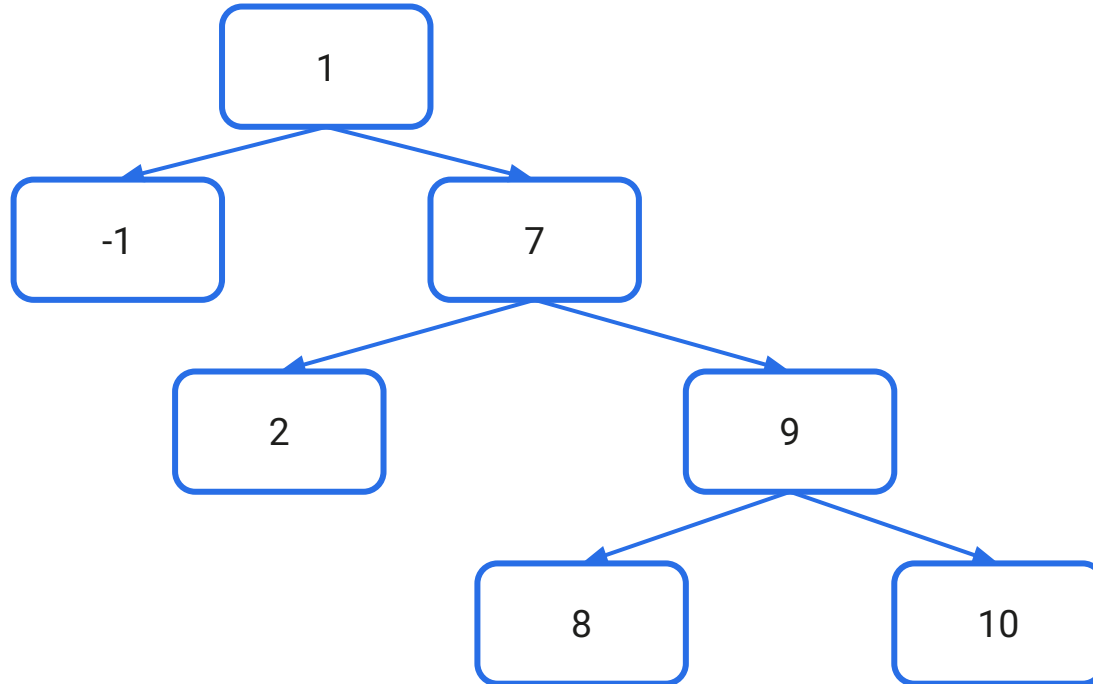


Quelle est la profondeur d'un arbre binaire équilibré en fonction du nombre de nœuds ?

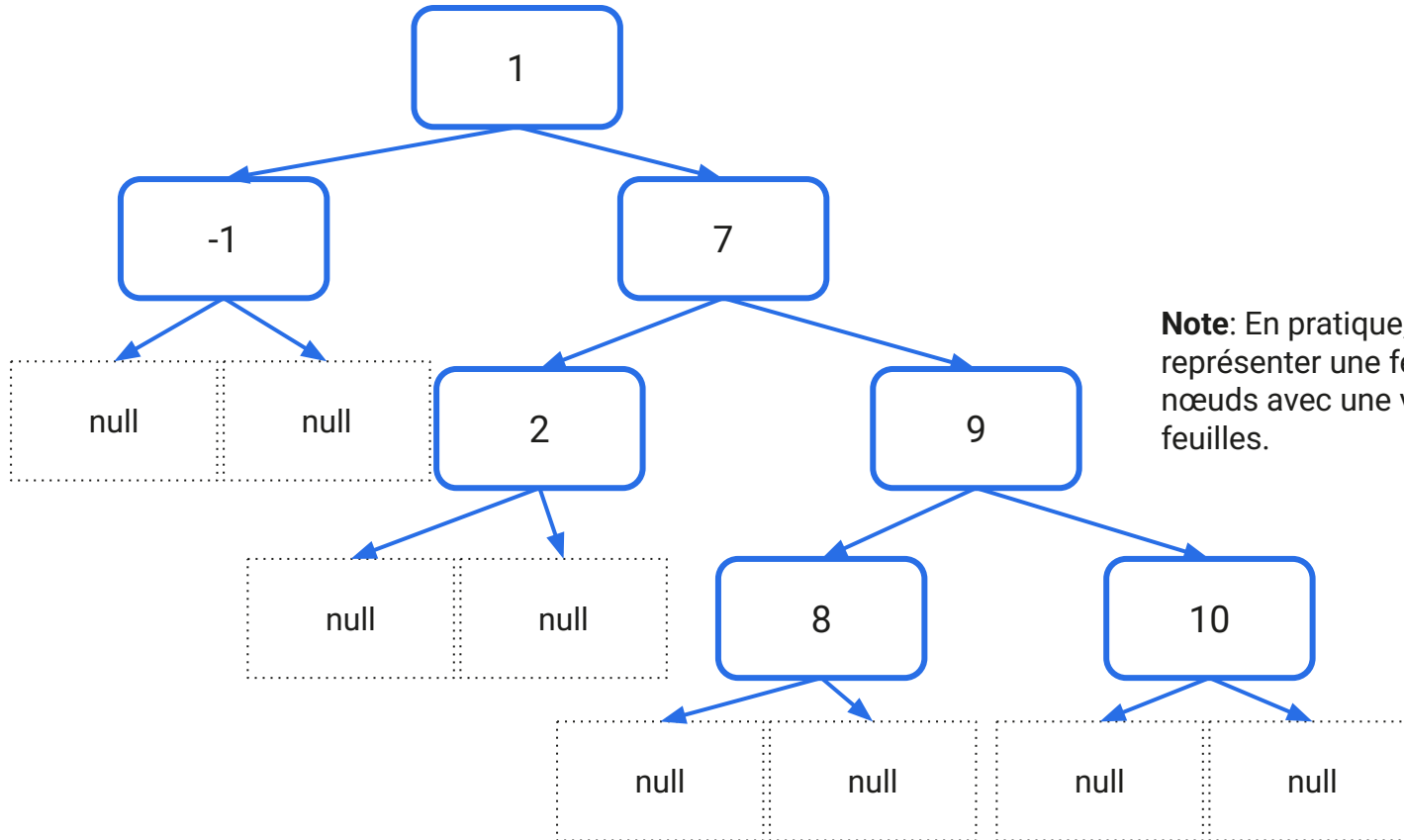
Elle vaut le logarithme du nombre de nœuds, arrondi à l'entier inférieur

Arbre binaire de recherche

- Un **arbre binaire de recherche** est un arbre binaire tel que, pour tout nœud N de l'arbre, tous les nœuds à gauche de N (son fils gauche et ses descendants) ont une valeur inférieure à N, et tous les nœuds à droite de N (son fils droit et ses descendants) ont une valeur supérieure à N.

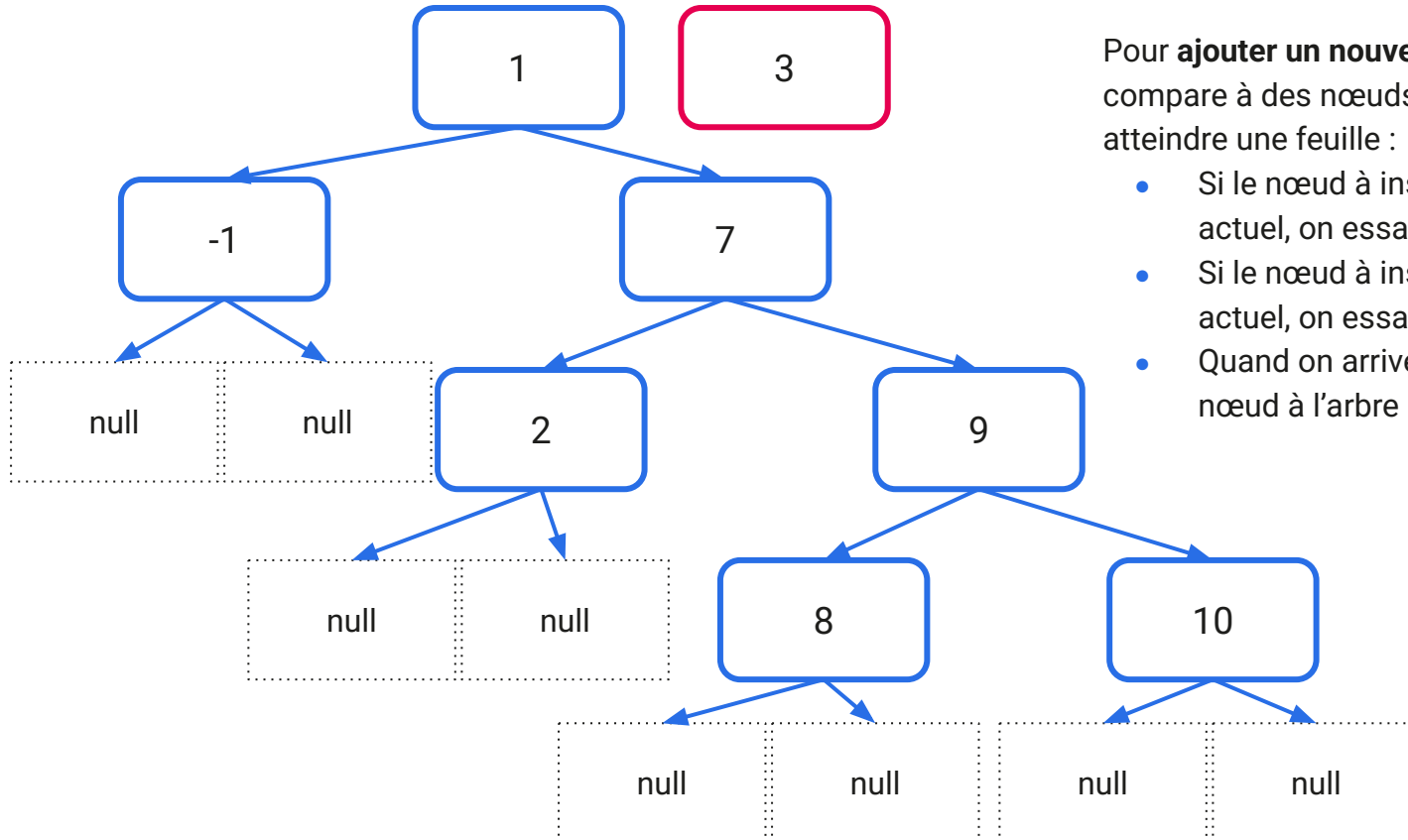


Arbre binaire de recherche



Note: En pratique, nous utilisons `null` pour représenter une feuille. Ainsi, tous les nœuds avec une valeur ne sont pas des feuilles.

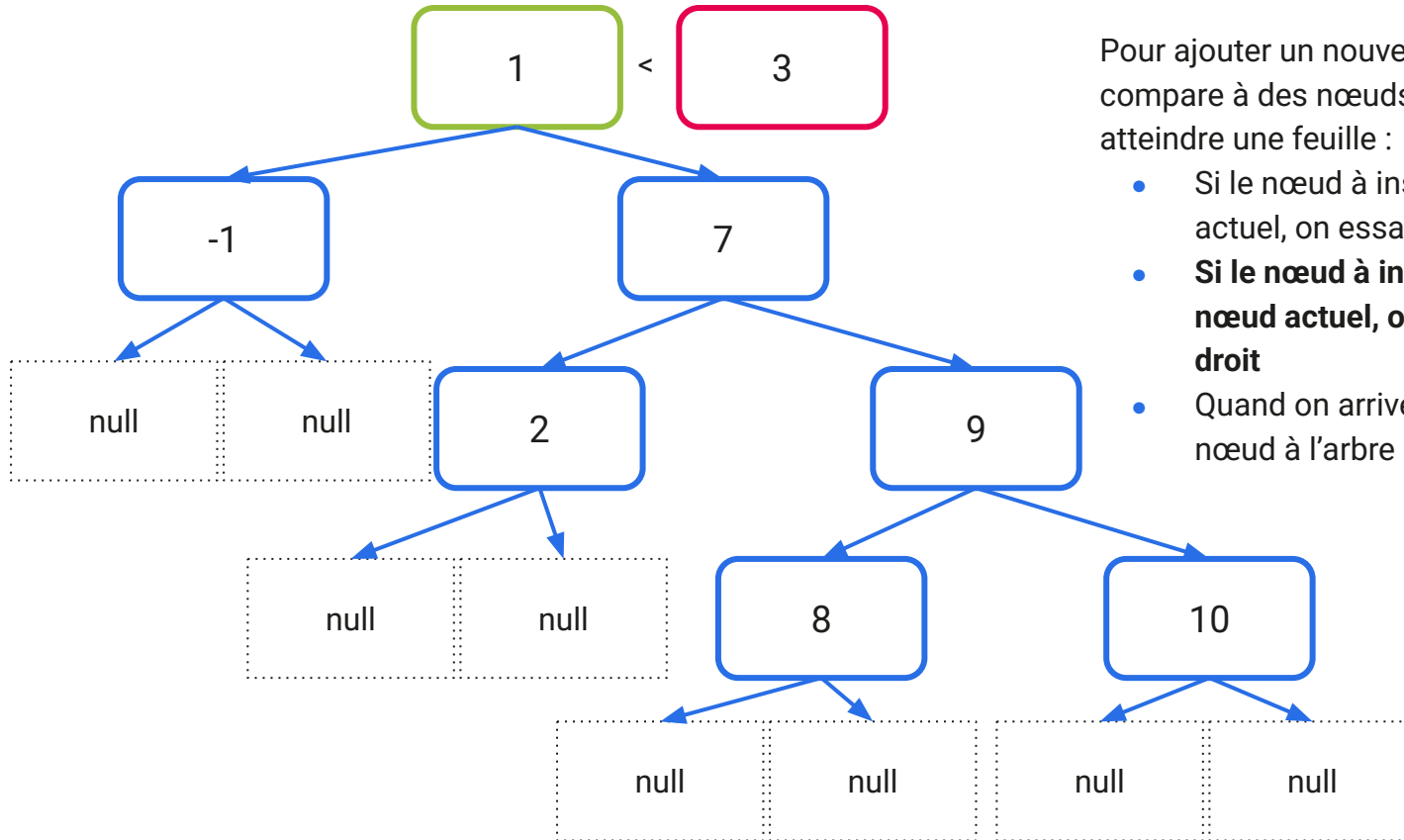
Arbre binaire de recherche - Ajout d'un élément



Pour **ajouter un nouveau nœud** dans l'arbre, on le compare à des nœuds en partant de la racine jusqu'à atteindre une feuille :

- Si le nœud à insérer est plus petit que le nœud actuel, on essaye de l'insérer du côté gauche
- Si le nœud à insérer est plus grand que le nœud actuel, on essaye de l'insérer du côté droit
- Quand on arrive à une feuille (null), on ajoute le nœud à l'arbre (à la place du null)

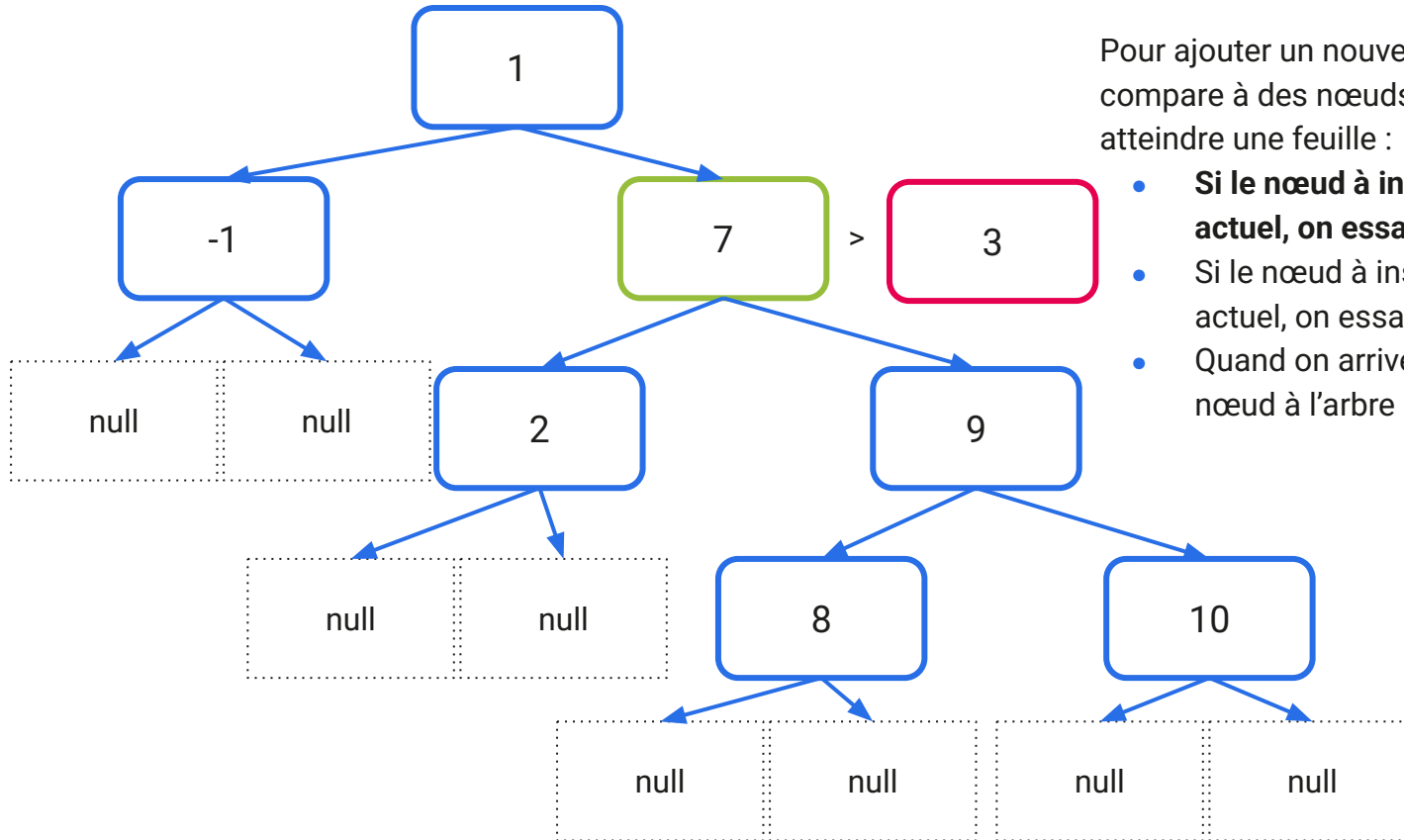
Arbre binaire de recherche - Ajout d'un élément



Pour ajouter un nouveau nœud dans l'arbre, on le compare à des nœuds en partant de la racine jusqu'à atteindre une feuille :

- Si le nœud à insérer est plus petit que le nœud actuel, on essaye de l'insérer du côté gauche
- **Si le nœud à insérer est plus grand que le nœud actuel, on essaye de l'insérer du côté droit**
- Quand on arrive à une feuille (null), on ajoute le nœud à l'arbre (à la place du null)

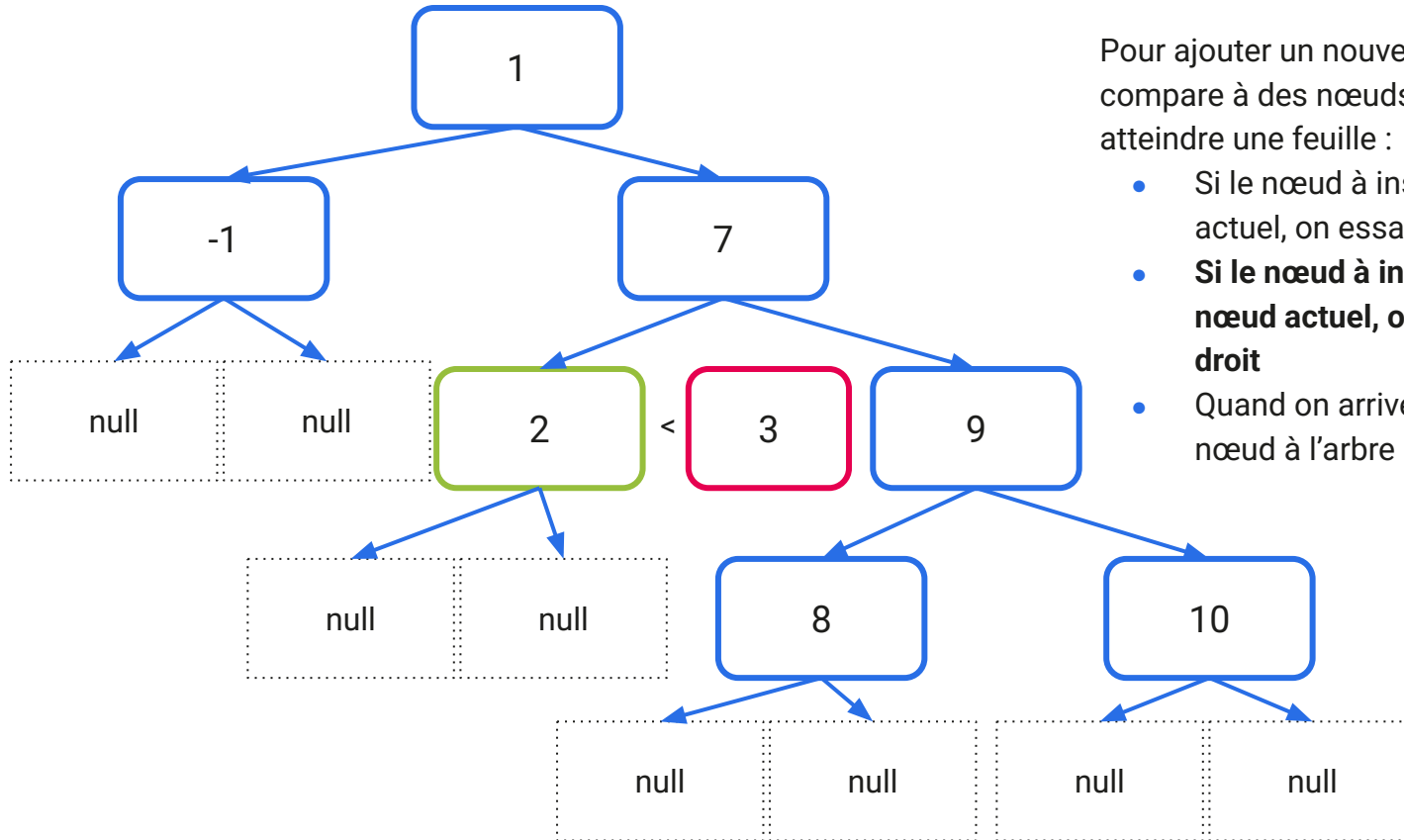
Arbre binaire de recherche - Ajout d'un élément



Pour ajouter un nouveau nœud dans l'arbre, on le compare à des nœuds en partant de la racine jusqu'à atteindre une feuille :

- Si le nœud à insérer est plus petit que le nœud actuel, on essaye de l'insérer du côté gauche
- Si le nœud à insérer est plus grand que le nœud actuel, on essaye de l'insérer du côté droit
- Quand on arrive à une feuille (null), on ajoute le nœud à l'arbre (à la place du null)

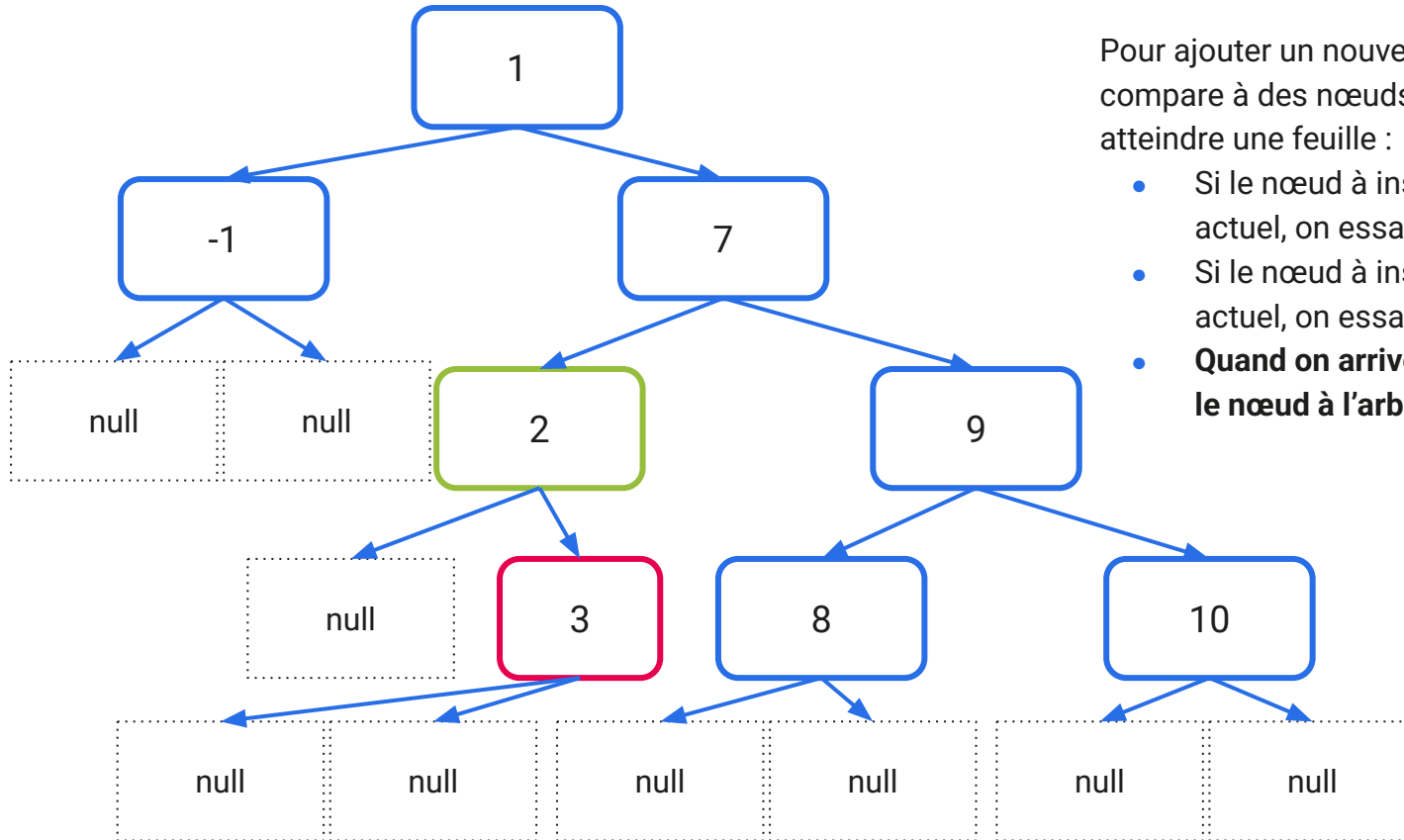
Arbre binaire de recherche - Ajout d'un élément



Pour ajouter un nouveau nœud dans l'arbre, on le compare à des nœuds en partant de la racine jusqu'à atteindre une feuille :

- Si le nœud à insérer est plus petit que le nœud actuel, on essaye de l'insérer du côté gauche
- **Si le nœud à insérer est plus grand que le nœud actuel, on essaye de l'insérer du côté droit**
- Quand on arrive à une feuille (null), on ajoute le nœud à l'arbre (à la place du null)

Arbre binaire de recherche - Ajout d'un élément



Pour ajouter un nouveau nœud dans l'arbre, on le compare à des nœuds en partant de la racine jusqu'à atteindre une feuille :

- Si le nœud à insérer est plus petit que le nœud actuel, on essaye de l'insérer du côté gauche
- Si le nœud à insérer est plus grand que le nœud actuel, on essaye de l'insérer du côté droit
- **Quand on arrive à une feuille (null), on ajoute le nœud à l'arbre (à la place du null)**

Arbre binaire de recherche - Recherche et suppression

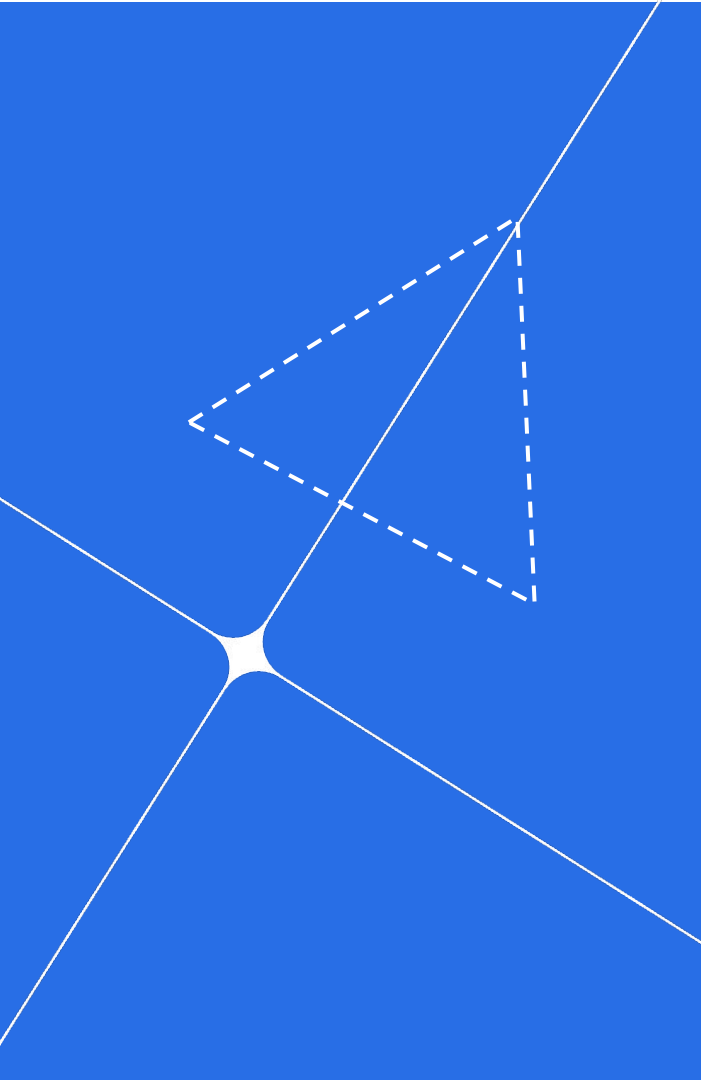
- La recherche d'un élément fonctionne de la même manière qu'un ajout: on descend l'arbre en partant de la racine et en allant à gauche ou à droite en fonction de la valeur recherchée, jusqu'à trouver l'élément (ou pas).
- La suppression est plus compliquée (voir les questions bonus à la fin)

Arbre binaire de recherche - Complexité

- Pour trouver les complexités, il va falloir réfléchir à la pire forme que pourrait avoir notre arbre par rapport à notre situation.
- Quel est le pire cas possible dans le cas d'un ajout ?

Arbre binaire de recherche - Complexité

- Pour trouver les complexités, il va falloir réfléchir à la pire forme que pourrait avoir notre arbre par rapport à notre situation.
- Quel est le pire cas possible dans le cas d'un ajout ?
 - On a un arbre filiforme et nous devons insérer le nouveau nœud tout en bas
 - Complexité linéaire en fonction du nombre de nœuds
- En pratique, la complexité sera logarithmique en moyenne (voir le TP pour en saisir l'intuition)



En route vers le TP !