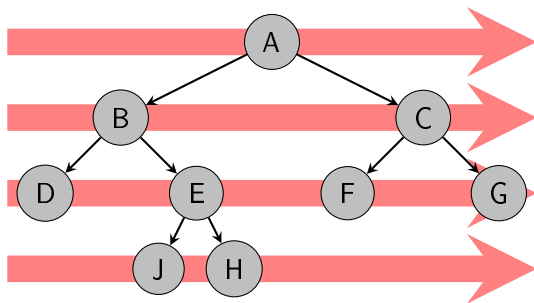


Quel ordre

Il s'agit de lire dans l'ordre lexicographique (ordre de lecture d'un texte, gauche à droite et de haut en bas)

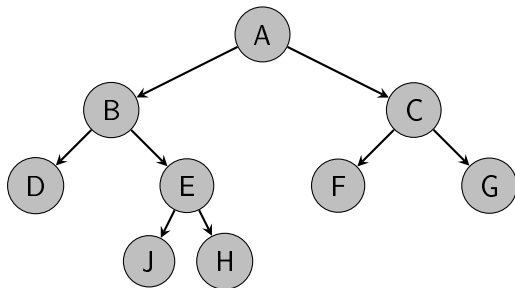


Dans cet exemple, on veut donc l'ordre A, B, C, D, E, F, G, J, H donc

Algorithme utilisant une file

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



Dans ce qui va suivre, nous allons détailler l'exécution de l'algorithme.

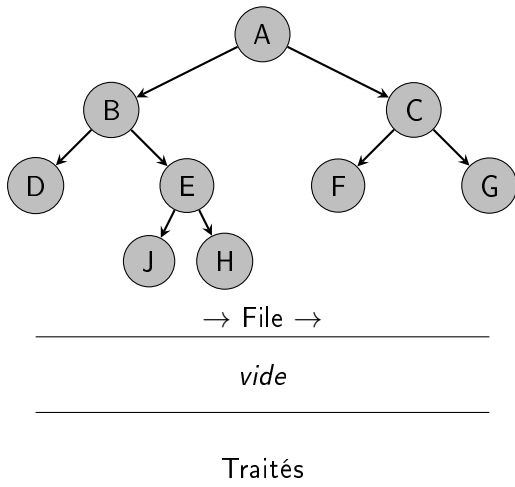
La **ligne verte** sera la ligne **venant d'être exécutée**.

La variable `nœud` dans l'algo désignera le nœud coloré en vert.

Algorithme utilisant une file - 1

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



Algorithme utilisant une file - 2

Entrées : arbre

1 Créer file vide

2 Enfiler racine de l'arbre

3 tant que *File non vide*

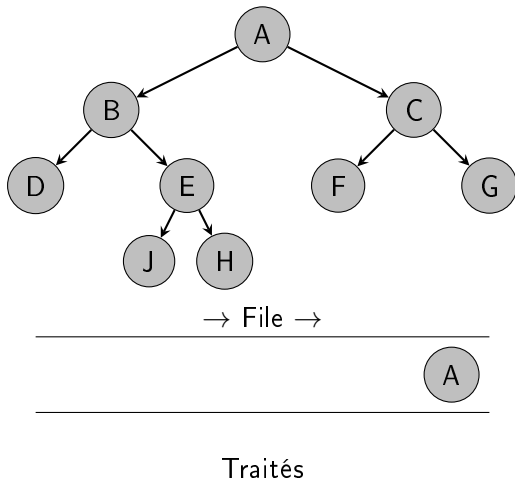
faire

4 | Défiler file → nœud

5 | Enfiler enfants

6 | TRAITEMENT(nœud)

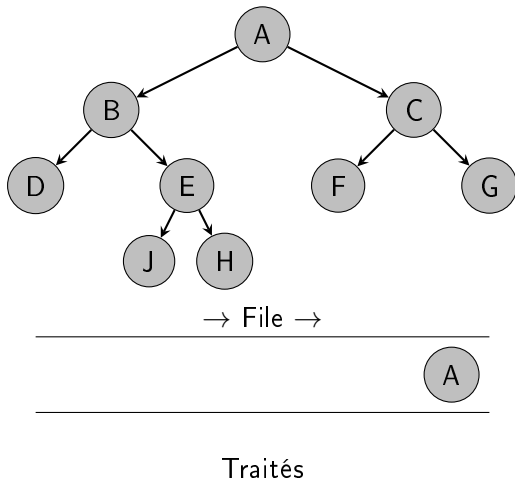
7 fin



Algorithme utilisant une file - 3

Entrées : arbre

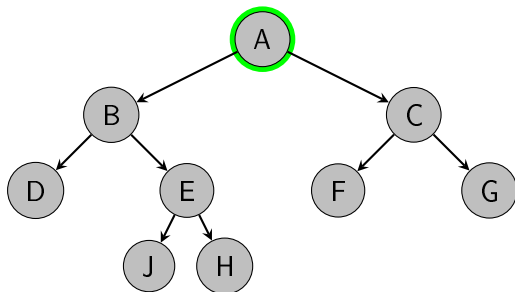
```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   | Défiler file → nœud
5   | Enfiler enfants
6   | TRAITEMENT(nœud)
7 fin
```



Algorithme utilisant une file - 4

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →

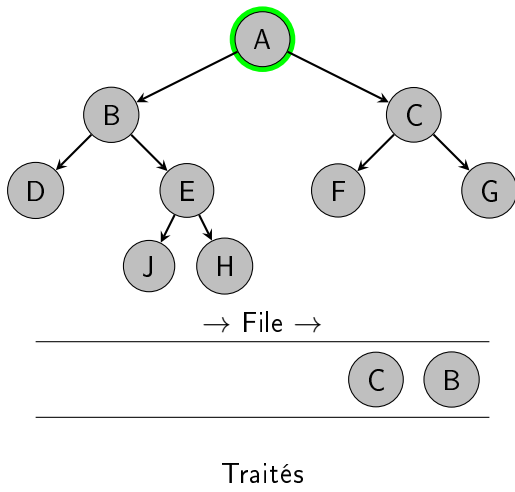
vide

Traités

Algorithme utilisant une file - 5

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   | Défiler file → nœud
5   | Enfiler enfants
6   | TRAITEMENT(nœud)
7 fin
```



Algorithme utilisant une file - 6

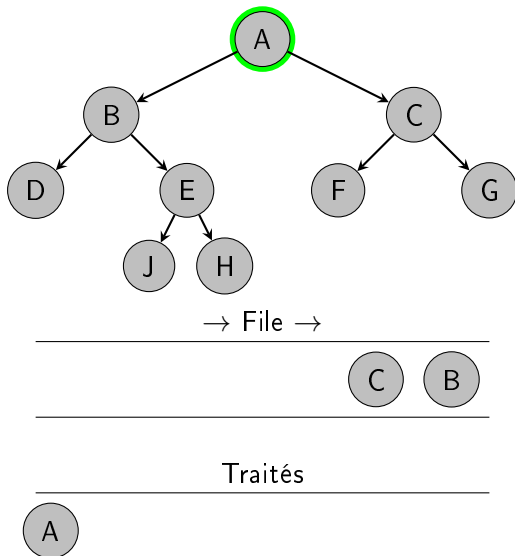
Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire

- 4 | Défiler file → nœud
- 5 | Enfiler enfants

- 6 | **TRAITEMENT**(nœud)

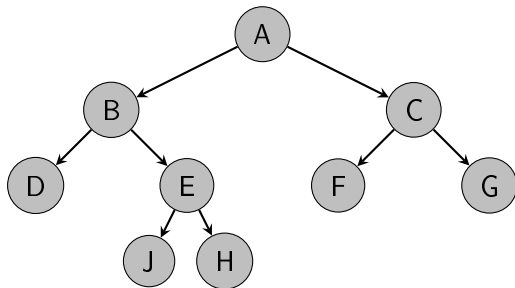
- 7 **fin**



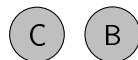
Algorithme utilisant une file - 7

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 Défiler file → nœud
 - 5 Enfiler enfants
 - 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →



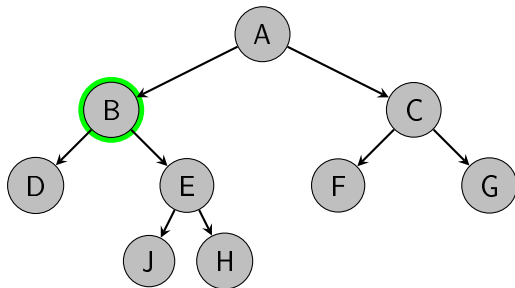
Traités



Algorithme utilisant une file - 8

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | Enfiler enfants
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



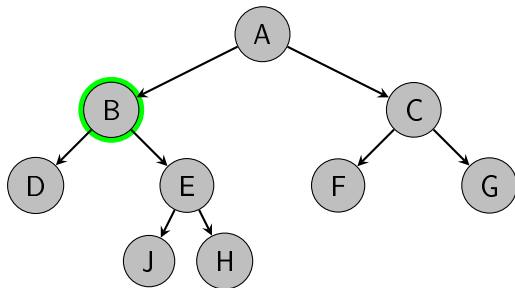
Traités



Algorithme utilisant une file - 9

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file $\rightarrow$ nœud
 - 5 | **Enfiler enfants**
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



$\rightarrow$ File $\rightarrow$



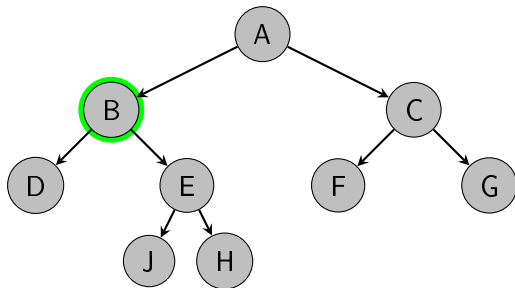
Traités



Algorithme utilisant une file - 10

Entrées : arbre

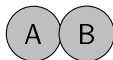
```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →



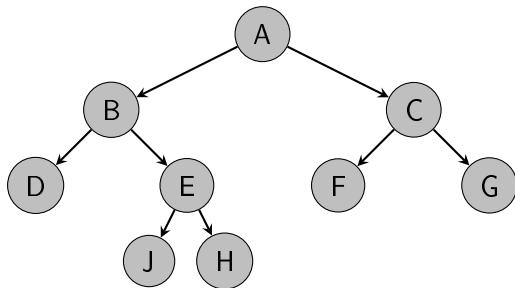
Traités



Algorithme utilisant une file - 11

Entrées : arbre

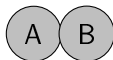
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | Enfiler enfants
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



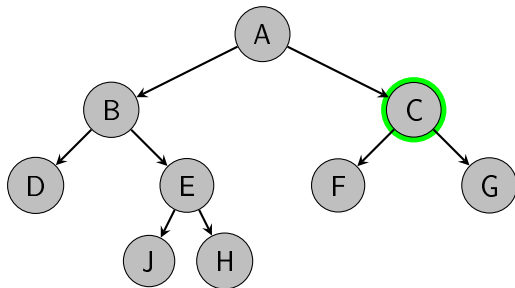
Traités



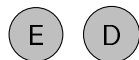
Algorithme utilisant une file - 12

Entrées : arbre

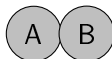
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- 4 **faire**
- 5 Défiler file → nœud
- 6 Enfiler enfants
- 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →



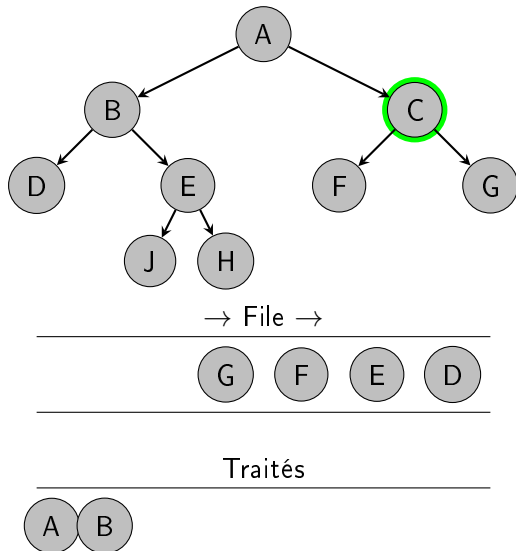
Traités



Algorithme utilisant une file - 13

Entrées : arbre

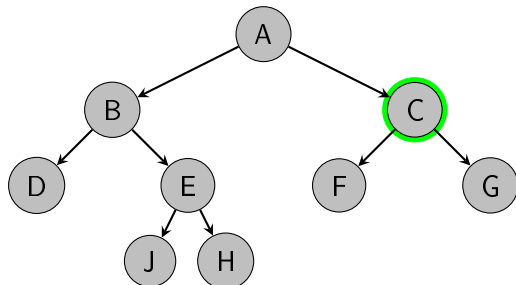
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | **Enfiler enfants**
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



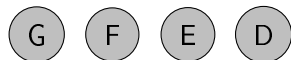
Algorithme utilisant une file - 14

Entrées : arbre

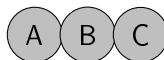
```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →



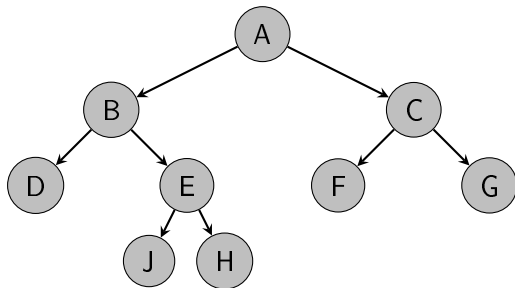
Traités



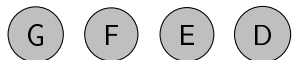
Algorithme utilisant une file - 15

Entrées : arbre

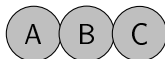
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | Enfiler enfants
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



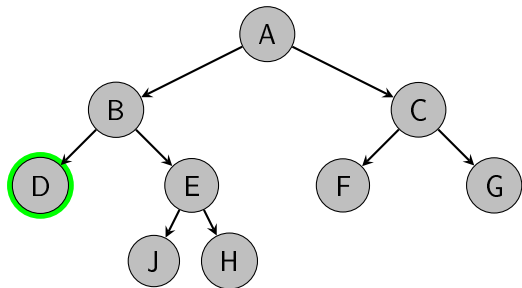
Traités



Algorithme utilisant une file - 16

Entrées : arbre

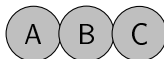
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- 4 **faire**
- 5 Défiler file → nœud
- 6 Enfiler enfants
- 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →



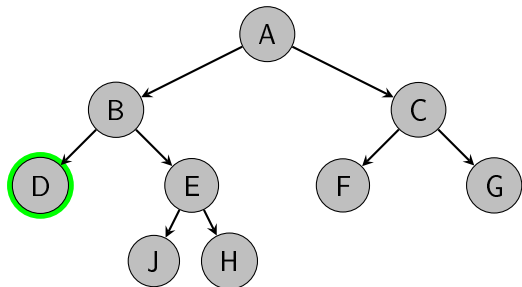
Traités



Algorithme utilisant une file - 17

Entrées : arbre

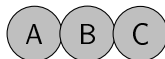
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | **Enfiler enfants**
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



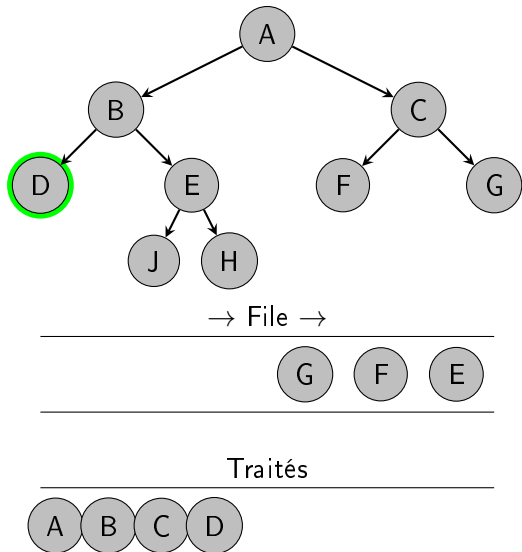
Traités



Algorithme utilisant une file - 18

Entrées : arbre

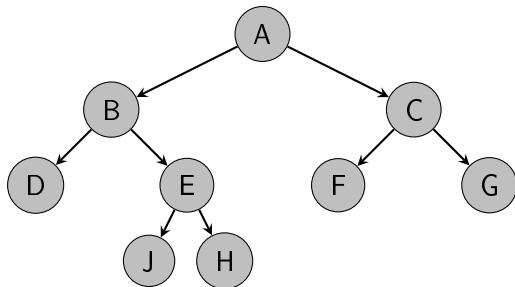
```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



Algorithme utilisant une file - 19

Entrées : arbre

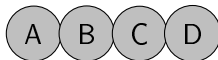
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 Défiler file → nœud
 - 5 Enfiler enfants
 - 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →



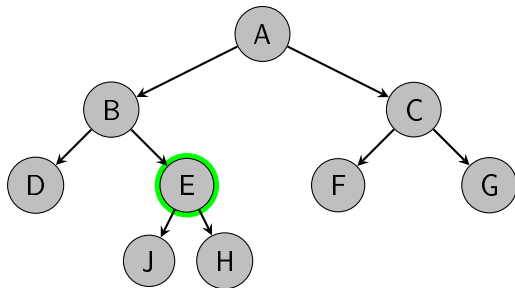
Traités



Algorithme utilisant une file - 20

Entrées : arbre

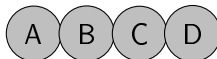
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 | Défiler file → nœud
- 5 | Enfiler enfants
- 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



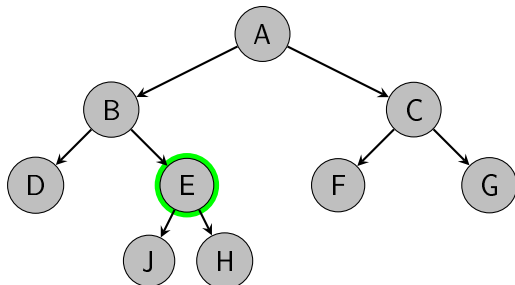
Traités



Algorithme utilisant une file - 21

Entrées : arbre

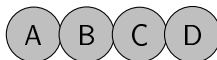
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file $\rightarrow$ nœud
 - 5 | **Enfiler enfants**
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



$\rightarrow$ File $\rightarrow$



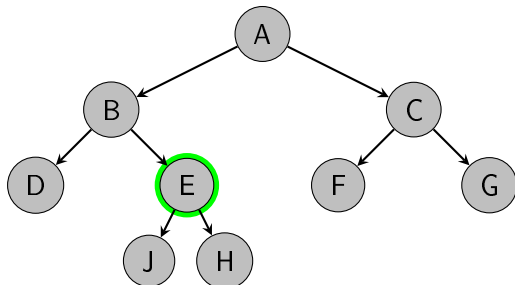
Traités



Algorithme utilisant une file - 22

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →



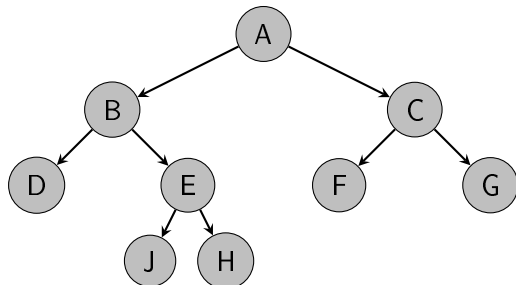
Traités



Algorithme utilisant une file - 23

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 Défiler file → nœud
 - 5 Enfiler enfants
 - 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →



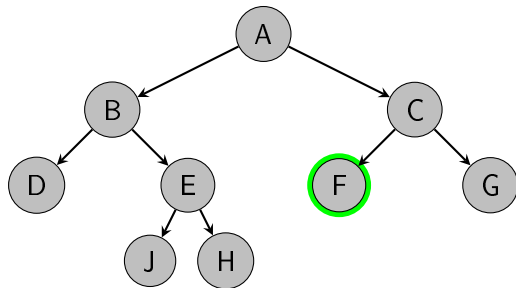
Traités



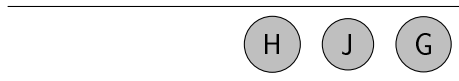
Algorithme utilisant une file - 24

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 | Défiler file → nœud
- 5 | Enfiler enfants
- 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



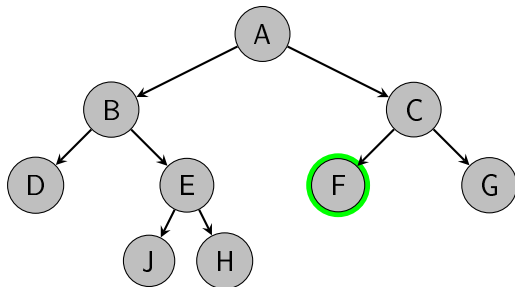
Traités



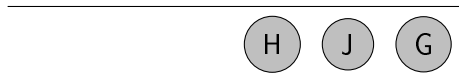
Algorithme utilisant une file - 25

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | **Enfiler enfants**
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



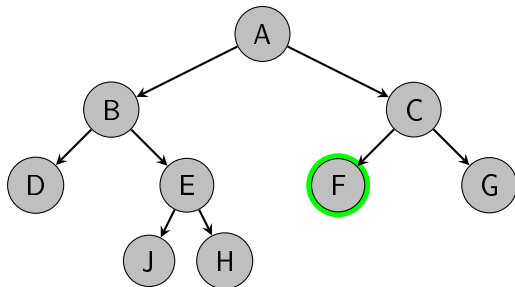
Traités



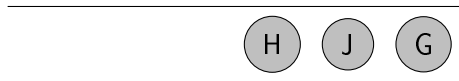
Algorithme utilisant une file - 26

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →



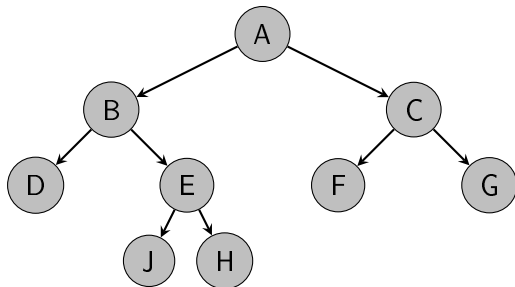
Traités



Algorithme utilisant une file - 27

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 Défiler file → nœud
 - 5 Enfiler enfants
 - 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →



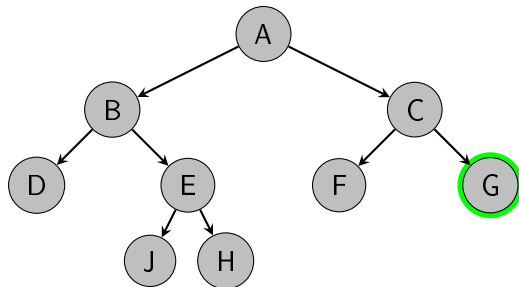
Traités



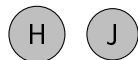
Algorithme utilisant une file - 28

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 | Défiler file → nœud
- 5 | Enfiler enfants
- 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



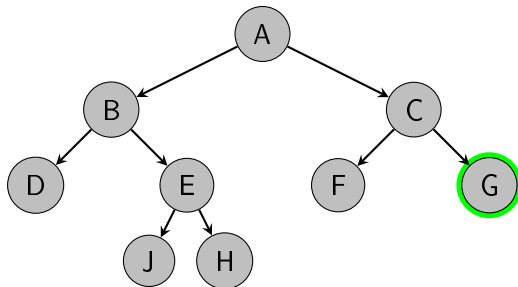
Traités



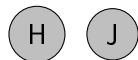
Algorithme utilisant une file - 29

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 | Défiler file → nœud
- 5 | **Enfiler enfants**
- 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



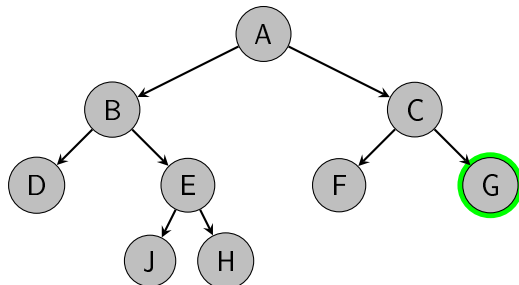
Traités



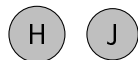
Algorithme utilisant une file - 30

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →



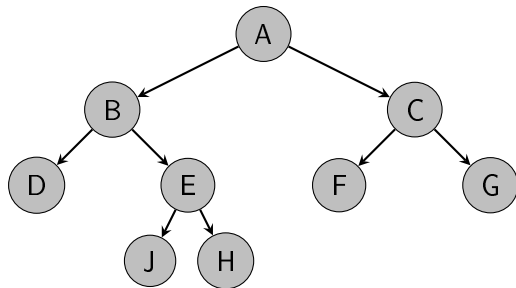
Traités



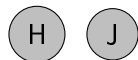
Algorithme utilisant une file - 31

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | Enfiler enfants
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



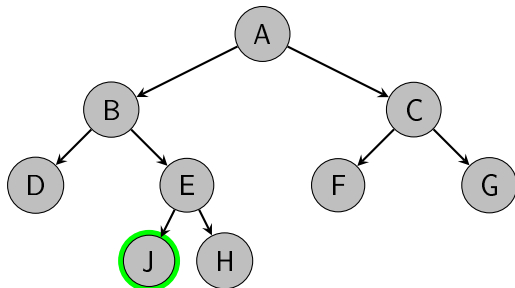
Traités



Algorithme utilisant une file - 32

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 | Défiler file → nœud
- 5 | Enfiler enfants
- 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →



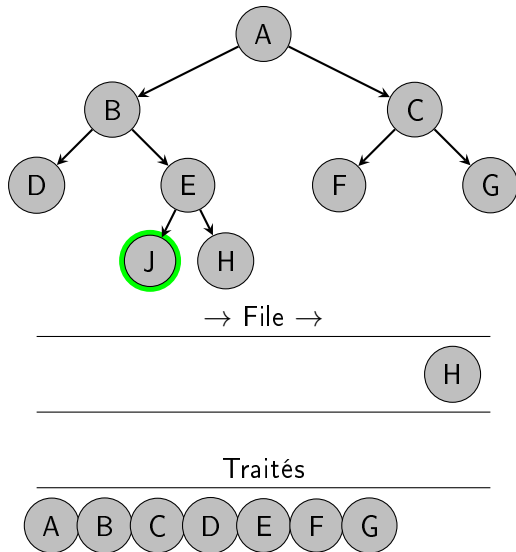
Traités



Algorithme utilisant une file - 33

Entrées : arbre

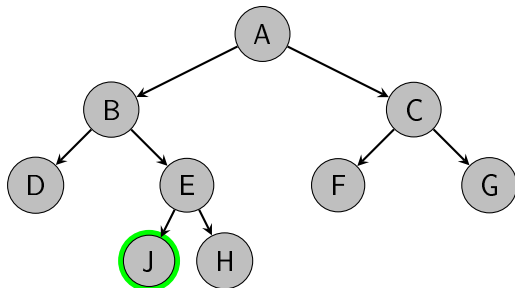
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 | Défiler file $\rightarrow$ nœud
- 5 | **Enfiler enfants**
- 6 | TRAITEMENT(nœud)
- 7 **fin**



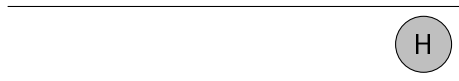
Algorithme utilisant une file - 34

Entrées : arbre

```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →



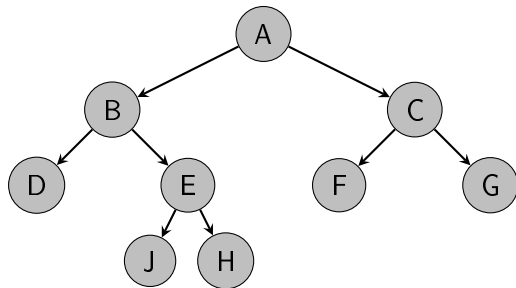
Traités



Algorithme utilisant une file - 35

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 Défiler file → nœud
 - 5 Enfiler enfants
 - 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →



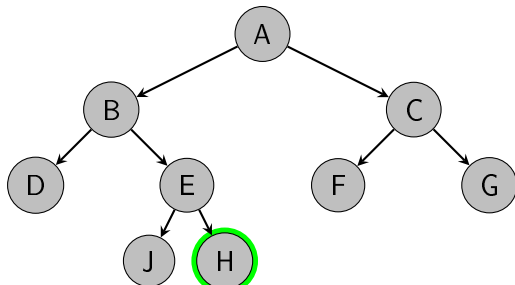
Traités



Algorithme utilisant une file - 36

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 | Défiler file → nœud
- 5 | Enfiler enfants
- 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →

vide

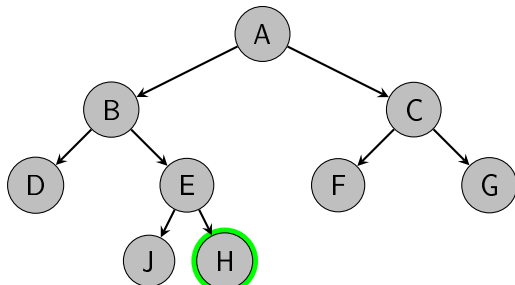
Traités



Algorithme utilisant une file - 37

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 | Défiler file → nœud
 - 5 | **Enfiler enfants**
 - 6 | TRAITEMENT(nœud)
- 7 **fin**



→ File →

vide

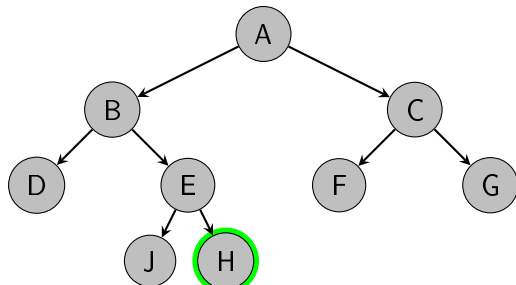
Traités



Algorithme utilisant une file - 38

Entrées : arbre

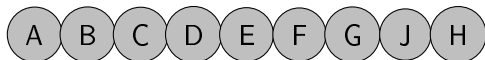
```
1 Créer file vide
2 Enfiler racine de l'arbre
3 tant que File non vide
  faire
4   Défiler file → nœud
5   Enfiler enfants
6   TRAITEMENT(nœud)
7 fin
```



→ File →

vide

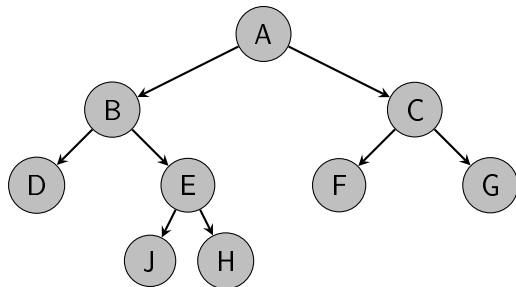
Traités



Algorithme utilisant une file - 39

Entrées : arbre

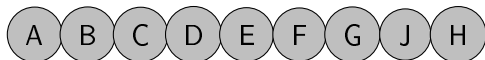
- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
 faire
 - 4 Défiler file → nœud
 - 5 Enfiler enfants
 - 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →

vide

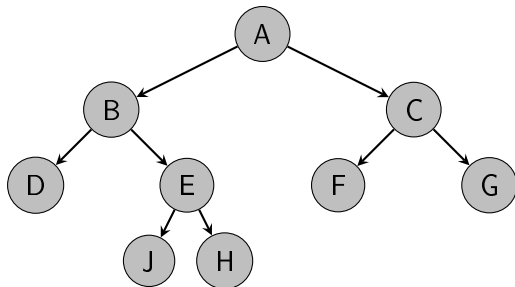
Traités



Algorithme utilisant une file - Fini !

Entrées : arbre

- 1 Créer file vide
- 2 Enfiler racine de l'arbre
- 3 **tant que** *File non vide*
- faire**
- 4 Défiler file → nœud
- 5 Enfiler enfants
- 6 TRAITEMENT(nœud)
- 7 **fin**



→ File →

vide

Traités

