

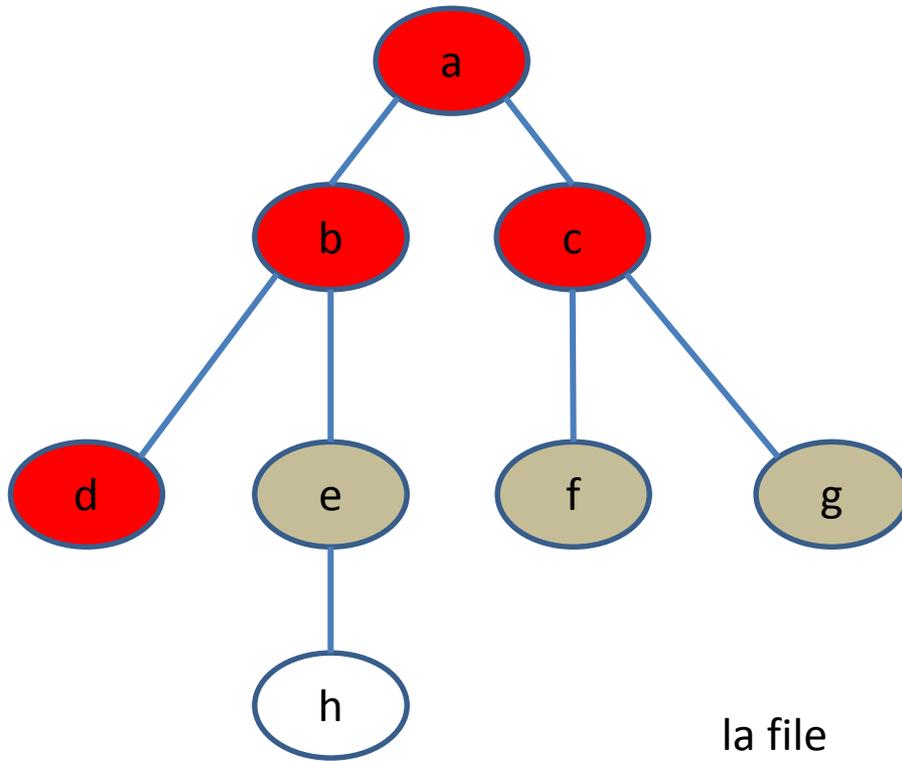
Parcours en largeur

L'**algorithme de parcours en largeur** (ou BFS, pour *Breadth First Search*) permet le parcours d'un graphe de manière itérative, en utilisant une file. Il peut par exemple servir à déterminer la connexité d'un graphe.

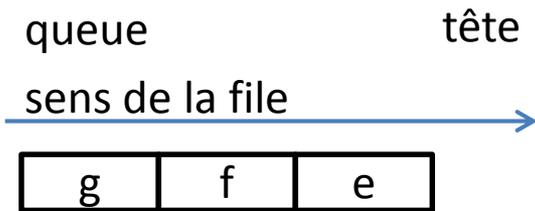
Étapes de l'algorithme :

- Mettre le nœud de départ dans la file.
- Retirer le nœud de la sortie (de la tête) de la file pour l'examiner.
- Mettre tous les voisins non examinés à l'entrée (à la queue) de la file.
- Si la file n'est pas vide reprendre à l'étape 2.

File: une structure de données du type *Premier entré, premier sorti*, en anglais FIFO (*First In, First Out*), ce qui veut dire que les premiers éléments ajoutés à la file seront les premiers à être récupérés, comme dans une file d'attente.

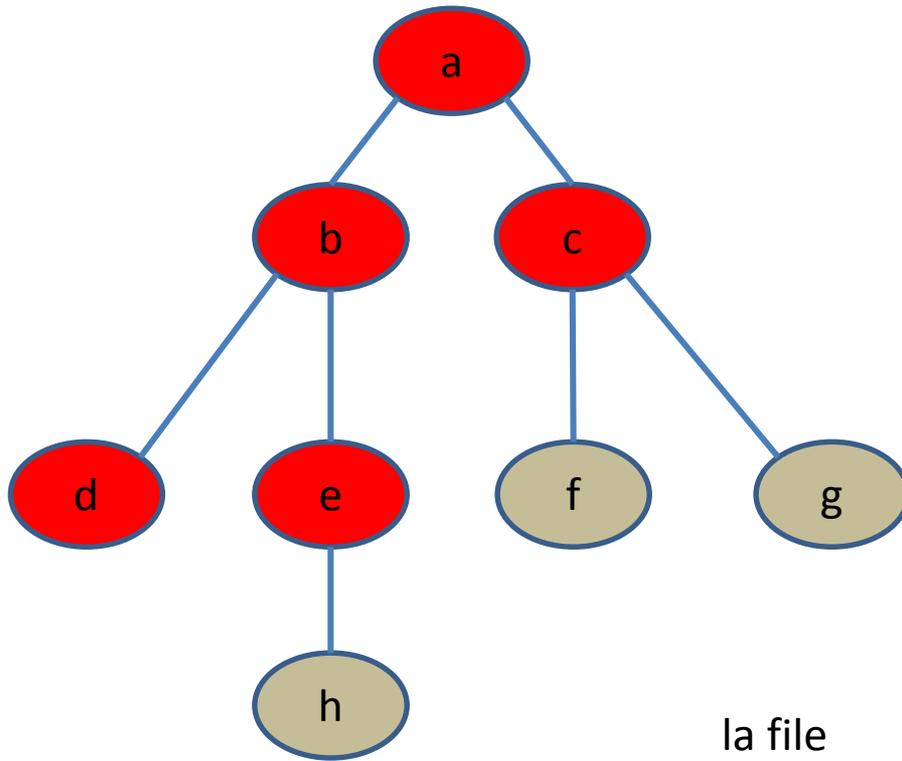


On met le sommet **d** dans la liste des sommets visités. Il n'a pas de voisins non visités.

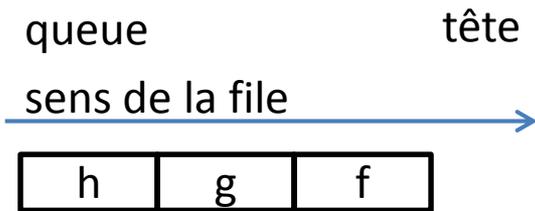


Sommets examinés



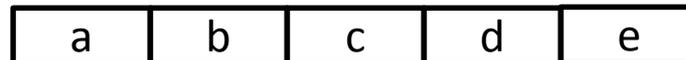


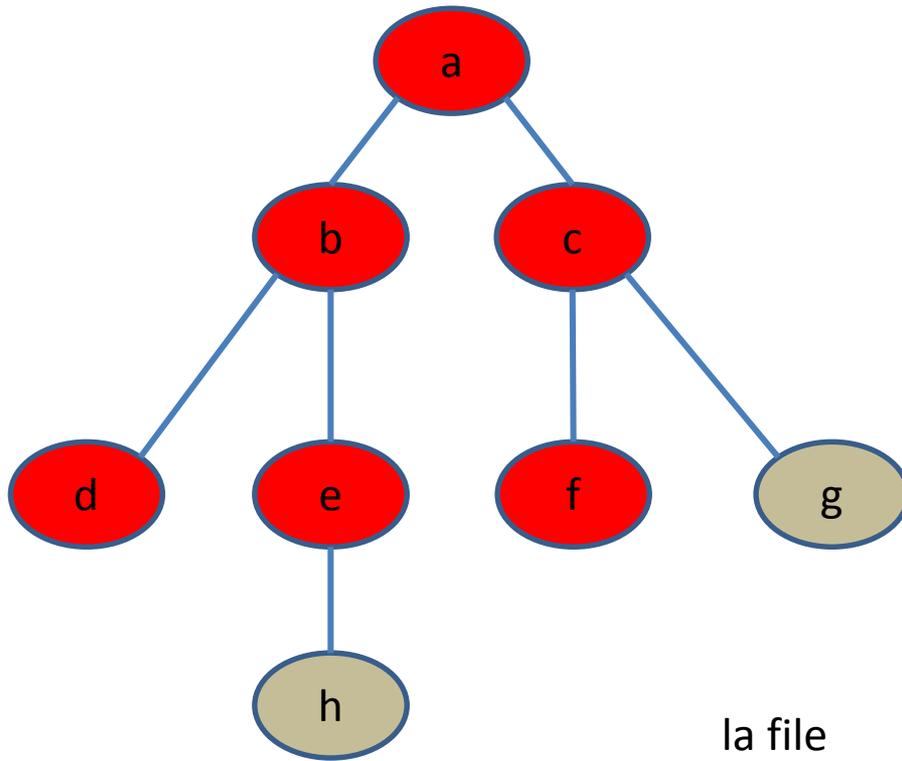
On met le sommet **e** dans la liste des sommets visités et on choisit son seul voisin non visités : **h**.



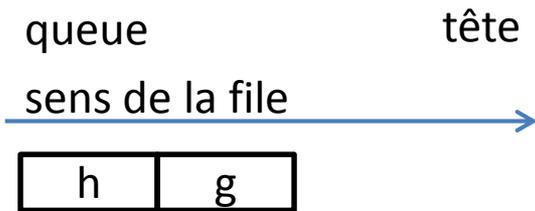
la file

Sommets examinés





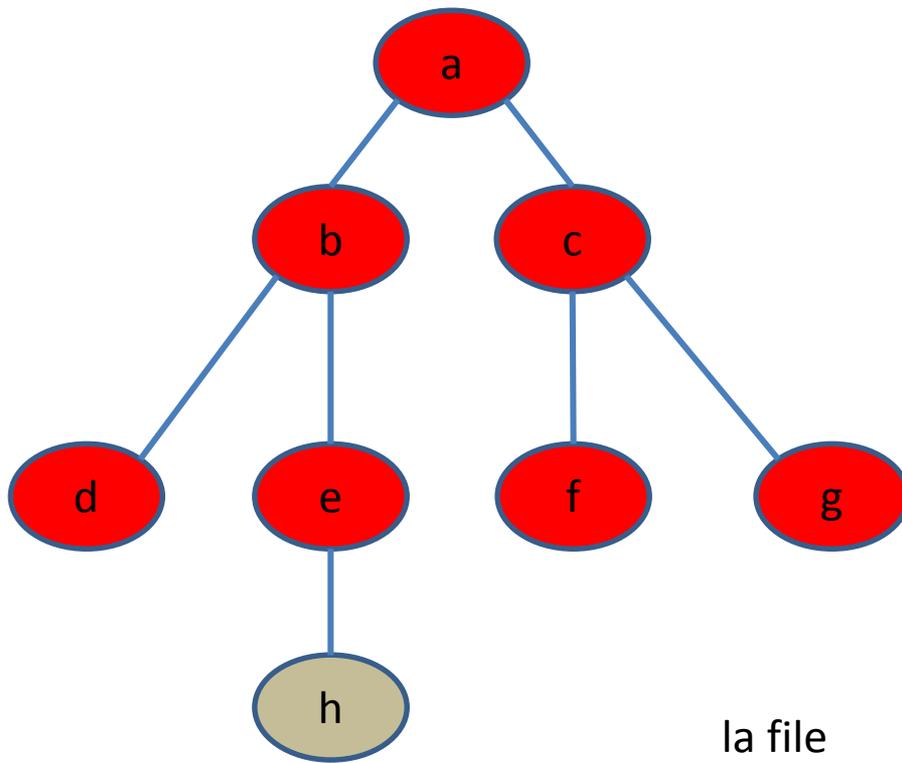
On met le sommet **f** dans la liste des sommets visités. Il n'a pas de voisins non visités.



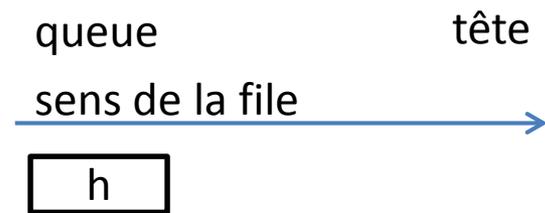
la file

Sommets examinés

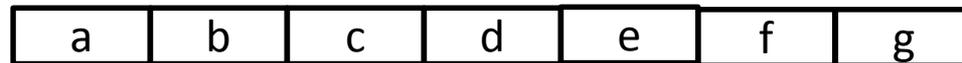


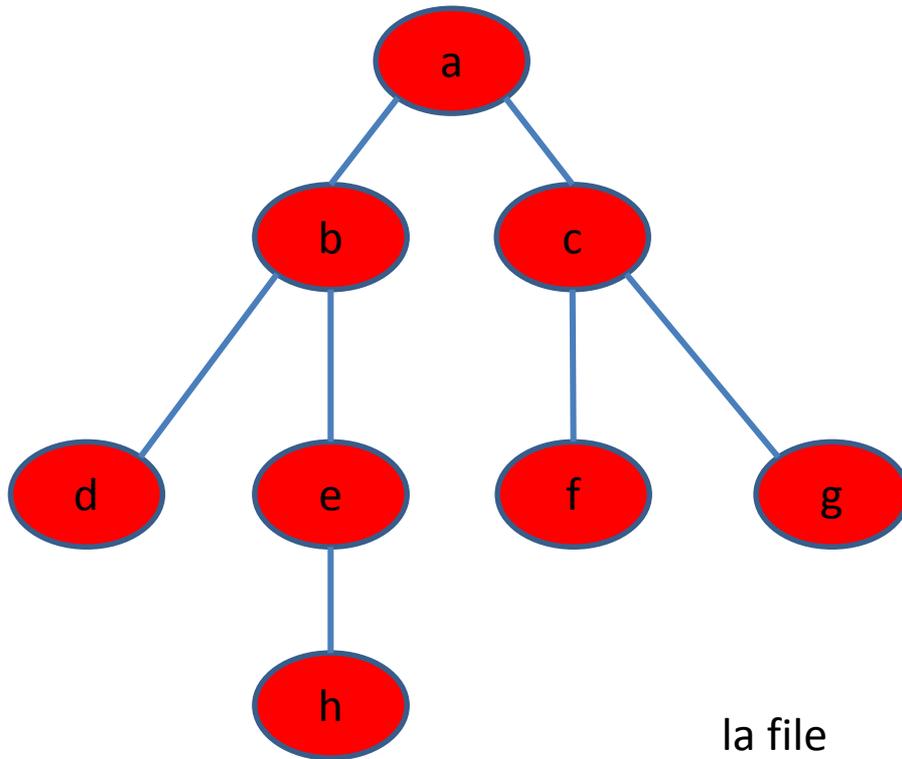


On met le sommet **g** dans la liste des sommets visités. Il n'a pas de voisins non visités.

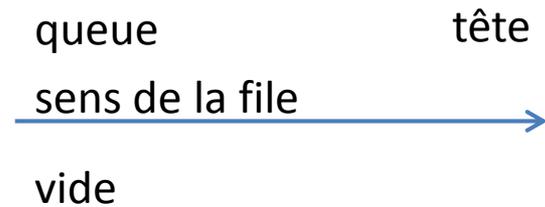


Sommets examinés





On met le sommet **h** dans la liste des sommets visités. Il n'a pas de voisins non visités.



la file

Sommets examinés

a	b	c	d	e	f	g	h
---	---	---	---	---	---	---	---

- **Exercice:** généraliser cet algorithme à des graphes contenant des cycles (qui ne sont pas des arbres)