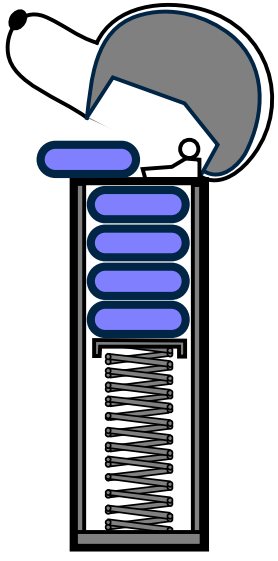
Introduction

Piles/files : structures conceptuelles



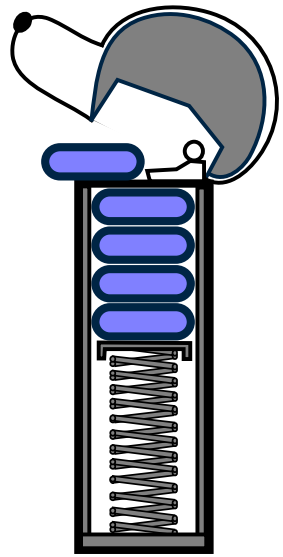
Des Piles, pourquoi ?

Quelques exemples d’utilisation :

- Recherche de chemins (labyrinthe)
- Gestion d’un historique de modifications (undo/redo)

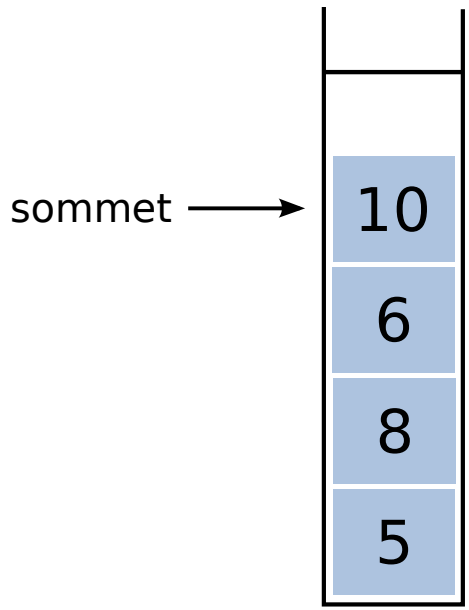
Pile

- Une pile est un conteneur d'éléments qui respecte le principe du *Last In First Out (LIFO) : le premier élément* accessible est le dernier ajouté.
- On ne peut accéder qu'à l'élément placé au sommet de la pile.
- Un distributeur de PEZ® en est un bon exemple :



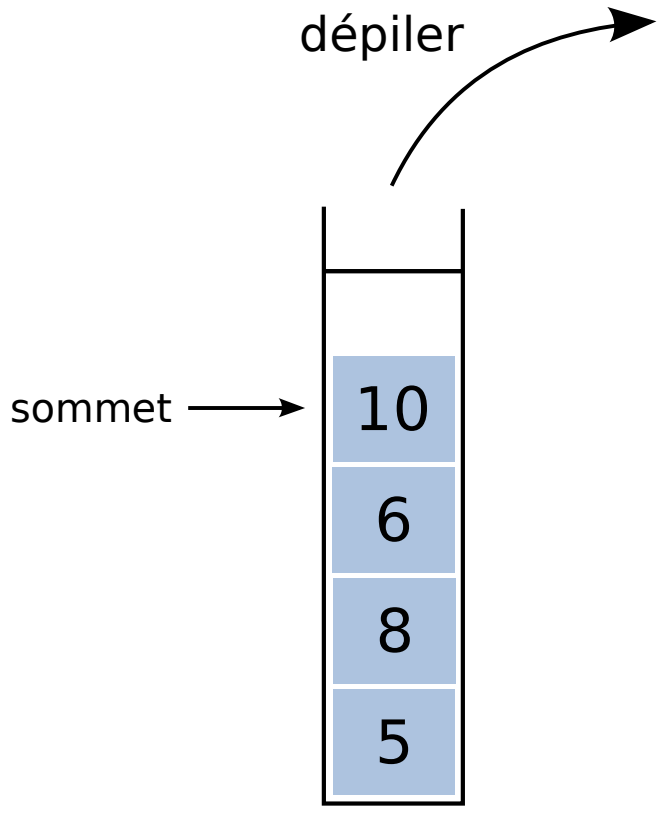
Pile : LIFO

Pile à l’origine, 10 est la valeur au sommet :



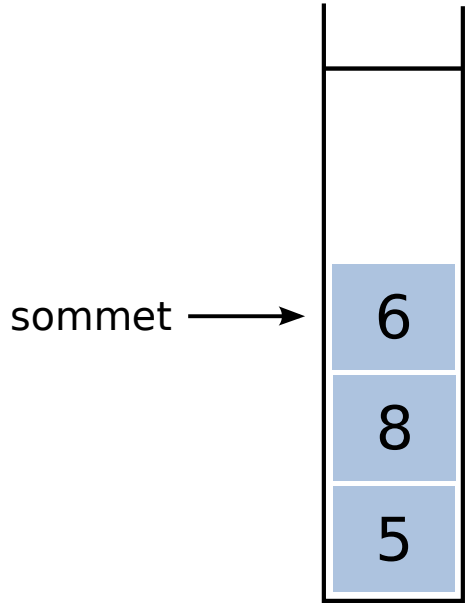
Pile : LIFO

On dépile 10 (retire la valeur 10 au sommet) :



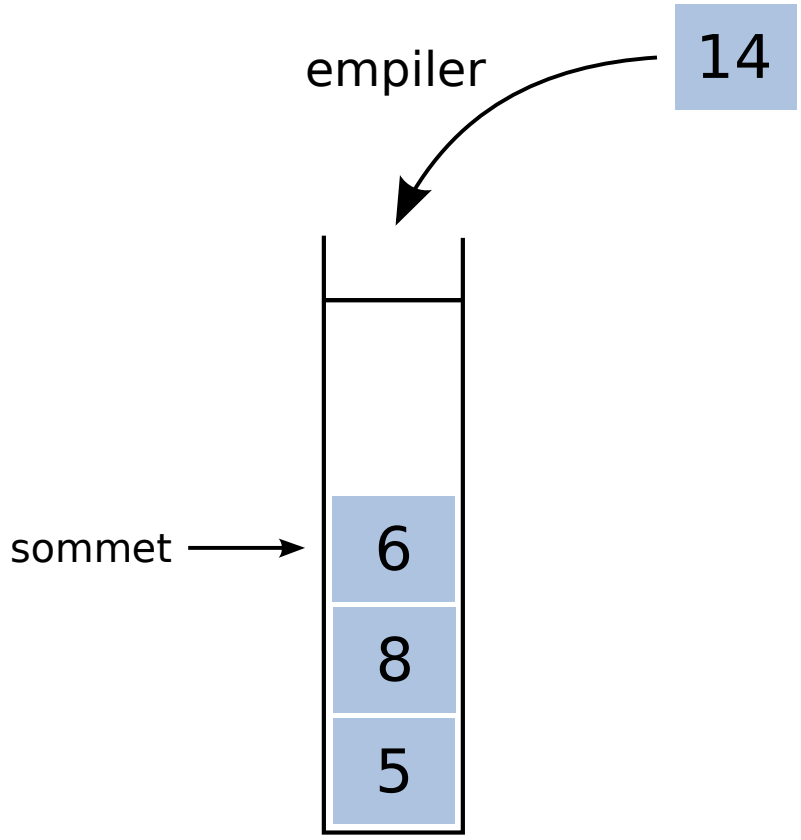
Pile : LIFO

6 devient la valeur au sommet de la pile :



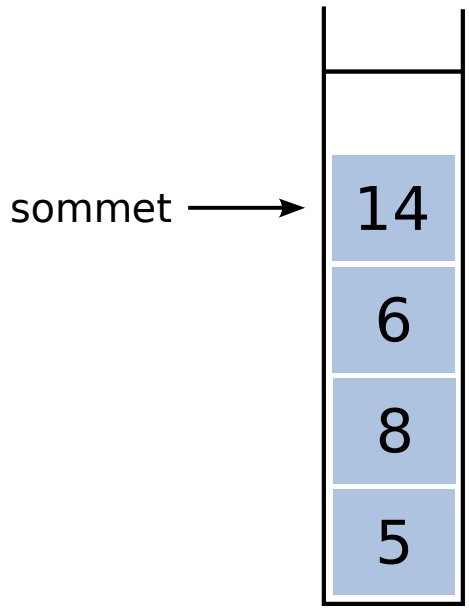
Pile : LIFO

On empile 14 (ajoute la valeur 14 au sommet) :



Pile : LIFO

Le sommet de la pile est 14 :



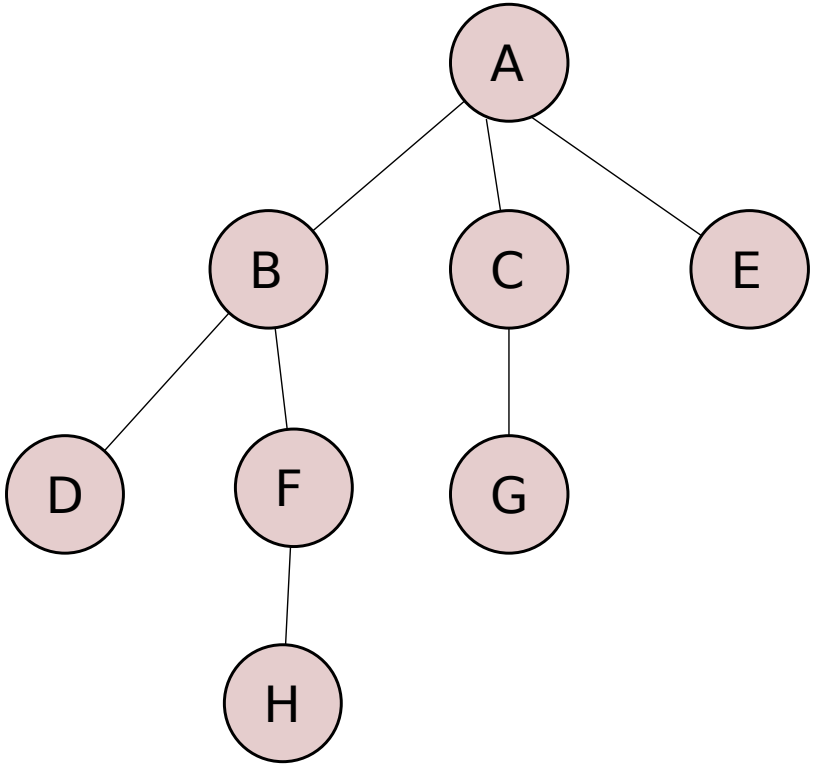
Pile

Opérations

Le type de données abstrait Pile spécifie plusieurs opérations appelées primitives :

Pile

Application : Parcours en profondeur

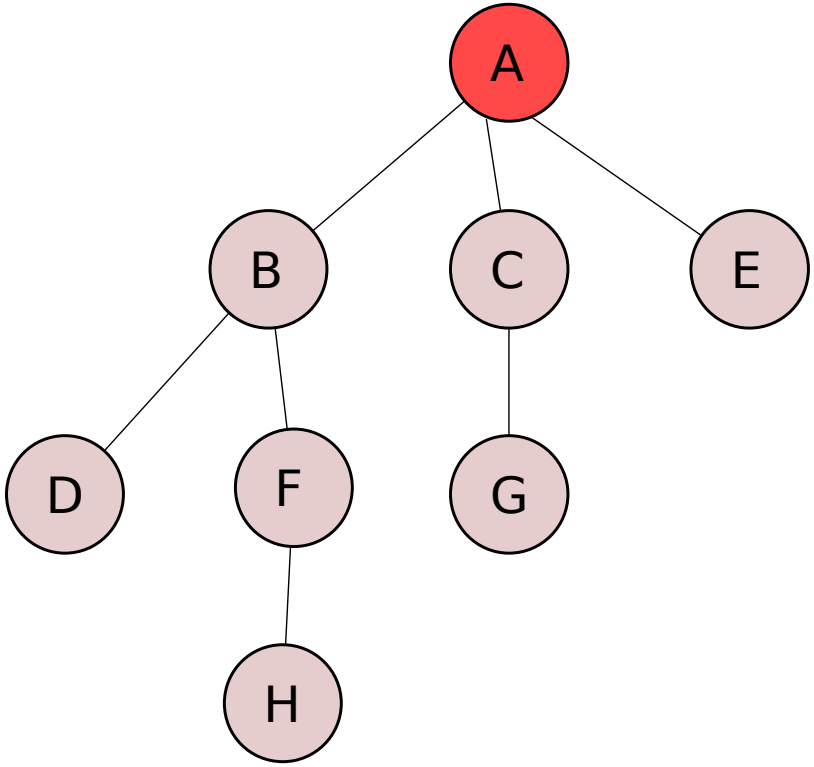


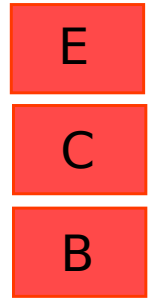
A

Ordre de visite : A

Pile

Application : Parcours en profondeur

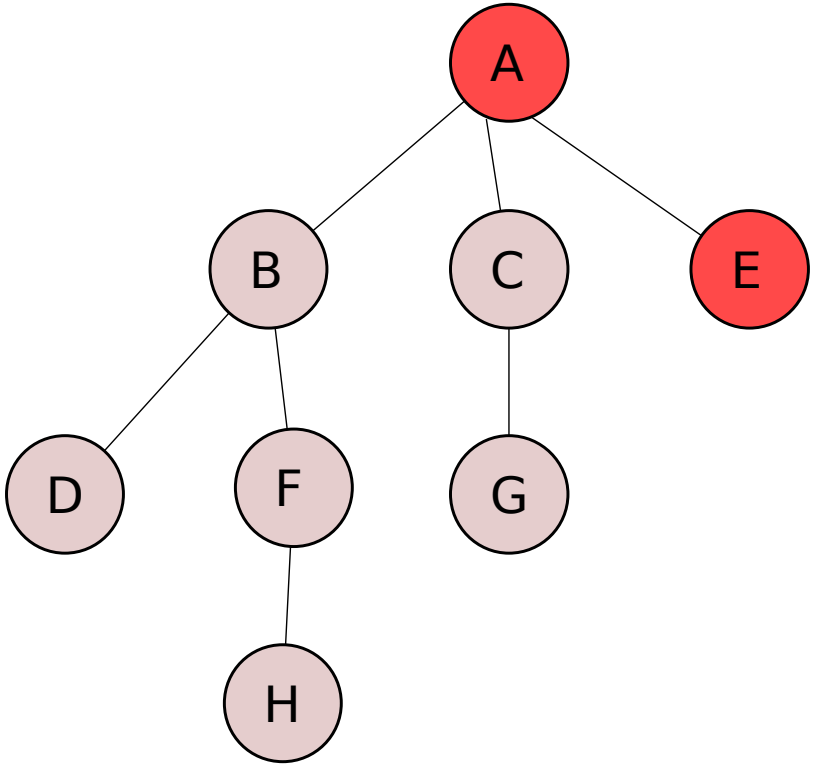


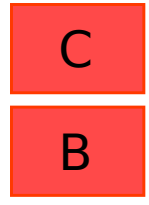


Ordre de visite : A-E

Pile

Application : Parcours en profondeur

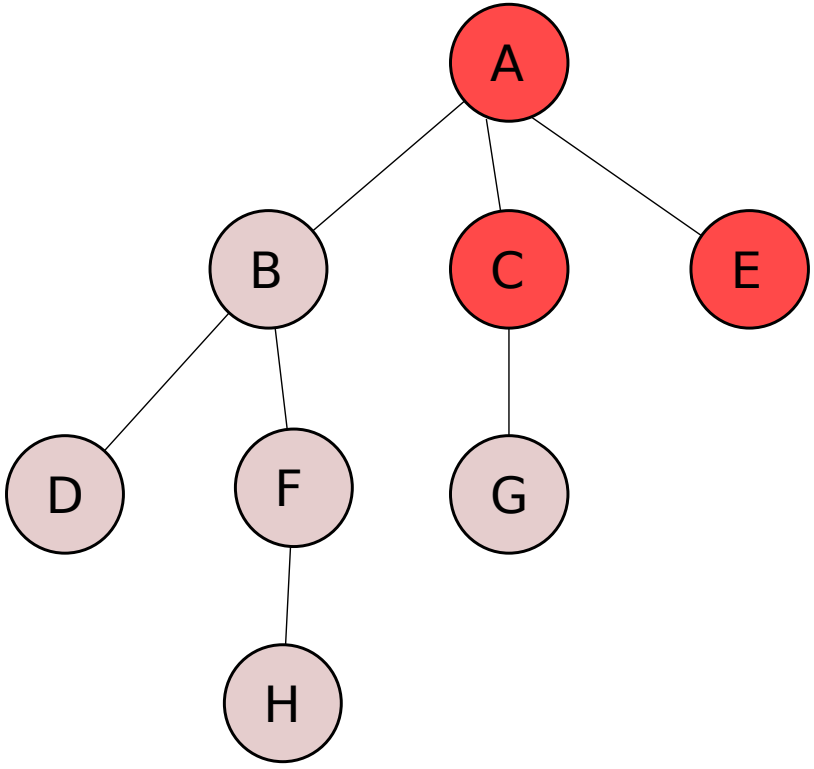


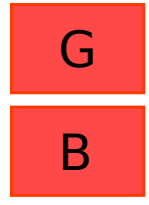


Ordre de visite : A-E-C

Pile

Application : Parcours en profondeur

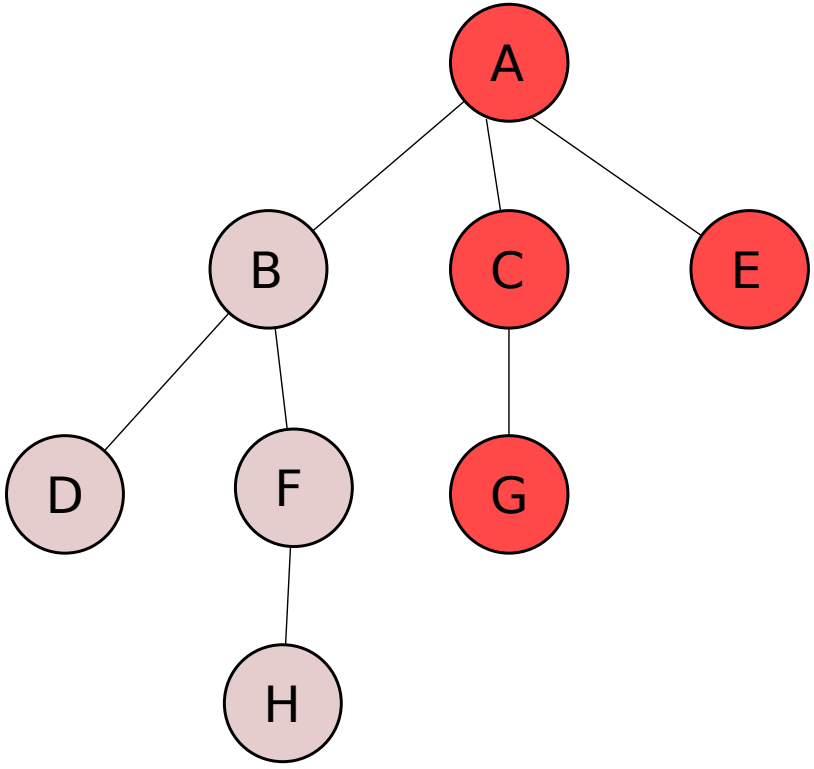




Ordre de visite : A-E-C-G

Pile

Application : Parcours en profondeur

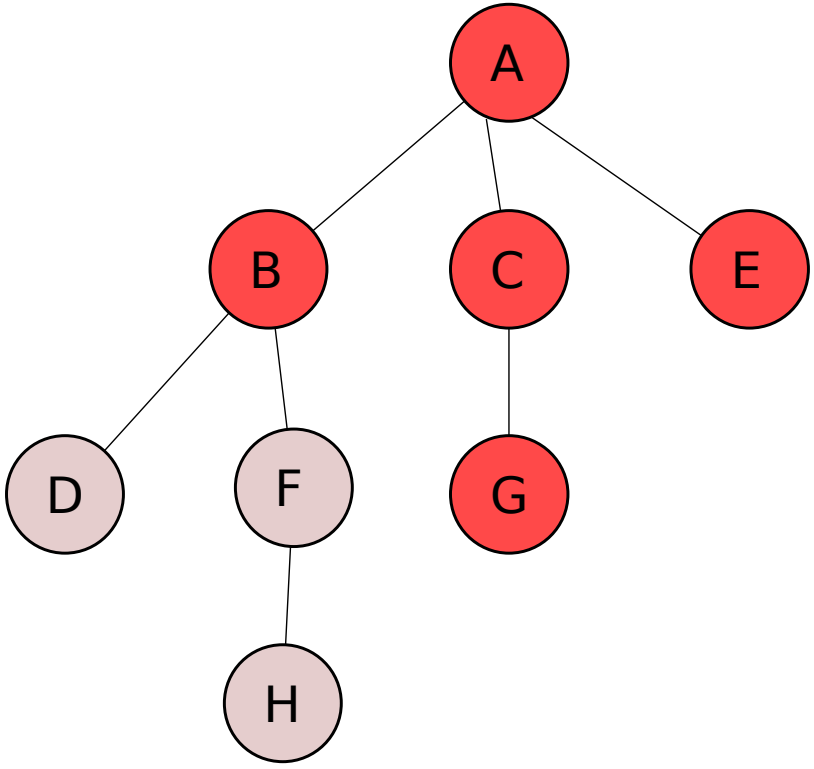


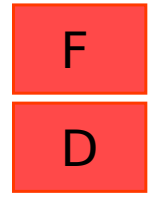
B

Ordre de visite : A-E-C-G-B

Pile

Application : Parcours en profondeur

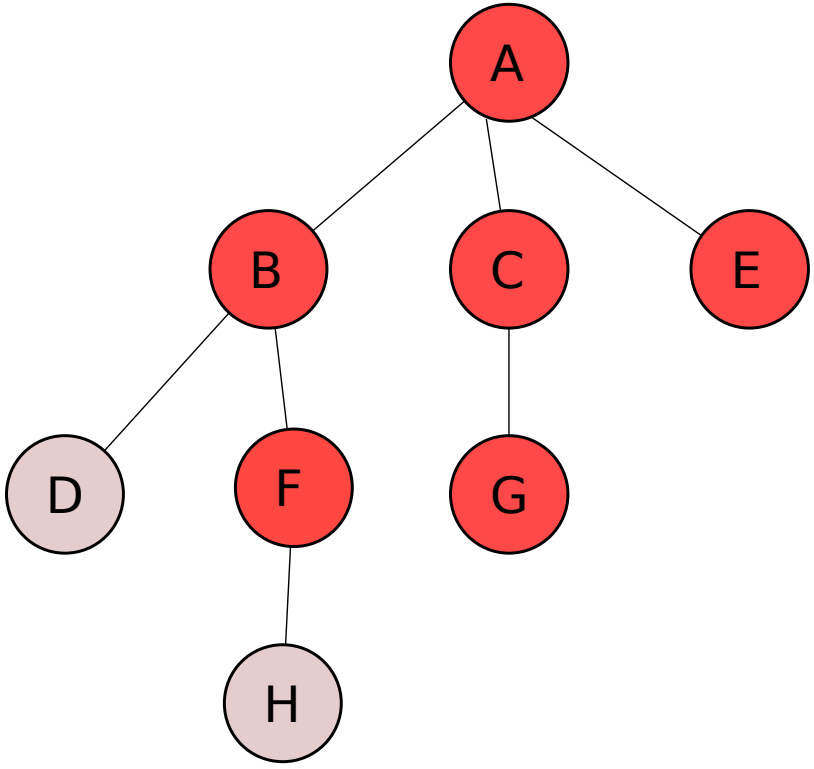


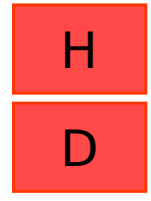


Ordre de visite : A-E-C-G-B-F

Pile

Application : Parcours en profondeur

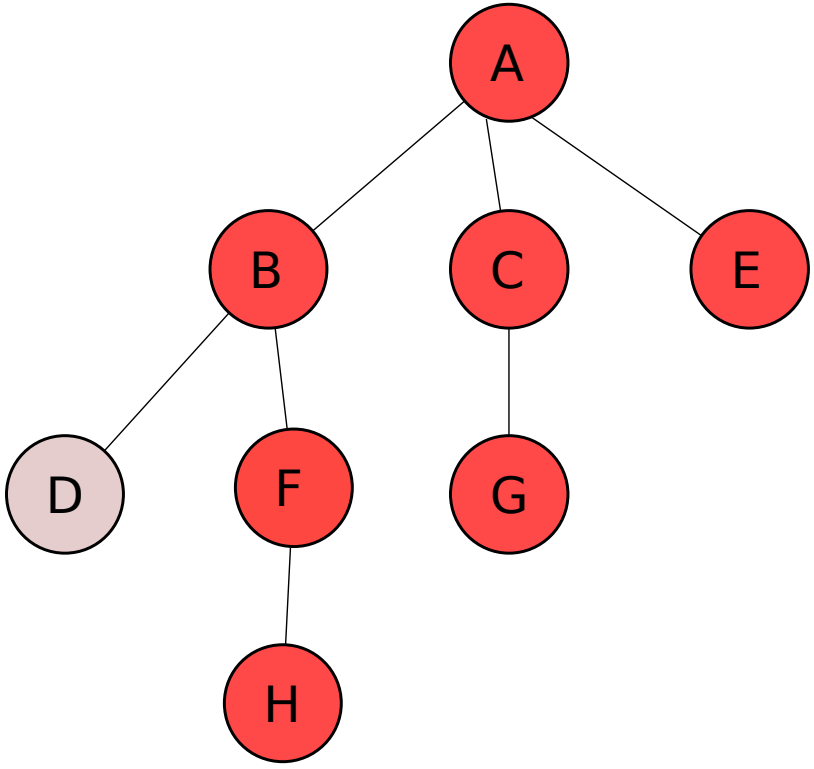




Ordre de visite : A-E-C-G-B-F-H

Pile

Application : Parcours en profondeur

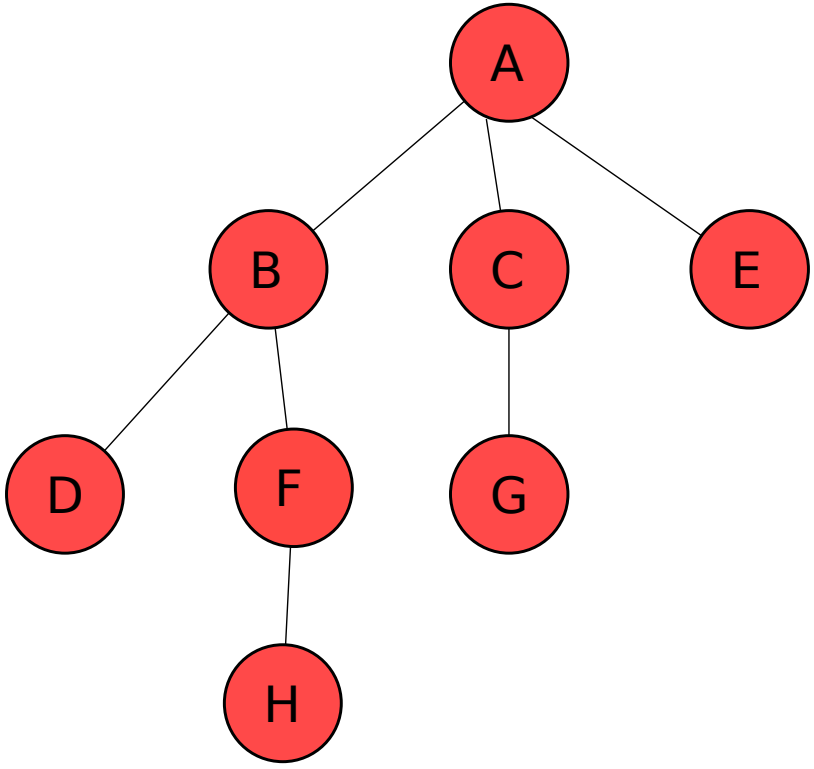


D

Ordre de visite : A-E-C-G-B-F-H-D

Pile

Application : Parcours en profondeur

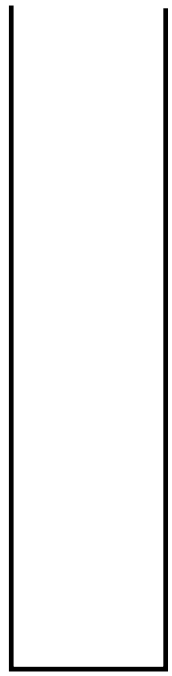


Ordre de visite : A-E-C-G-B-F-H-D

Exemple

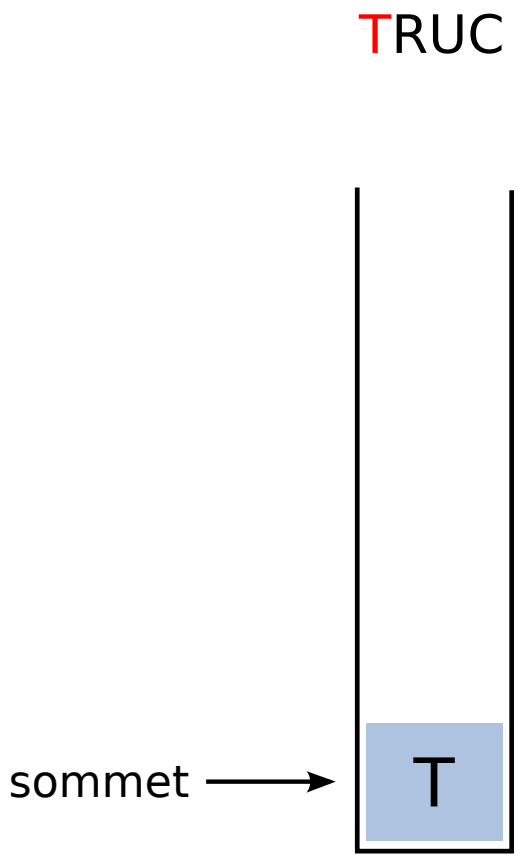
Inverser une chaîne de caractères

TRUC



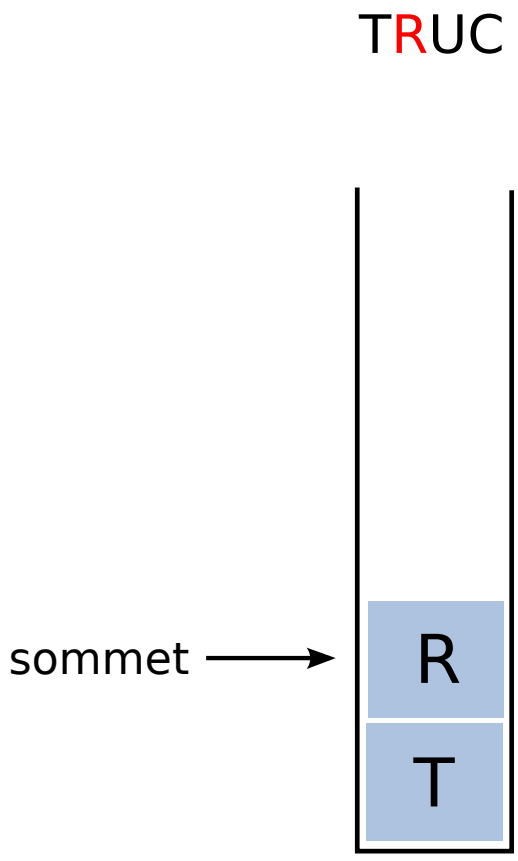
Exemple

Inverser une chaîne de caractères



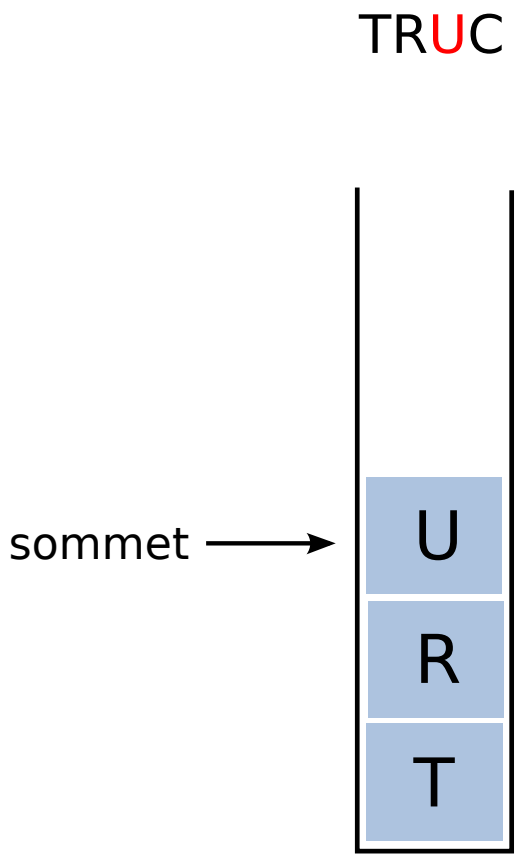
Exemple

Inverser une chaîne de caractères



Exemple

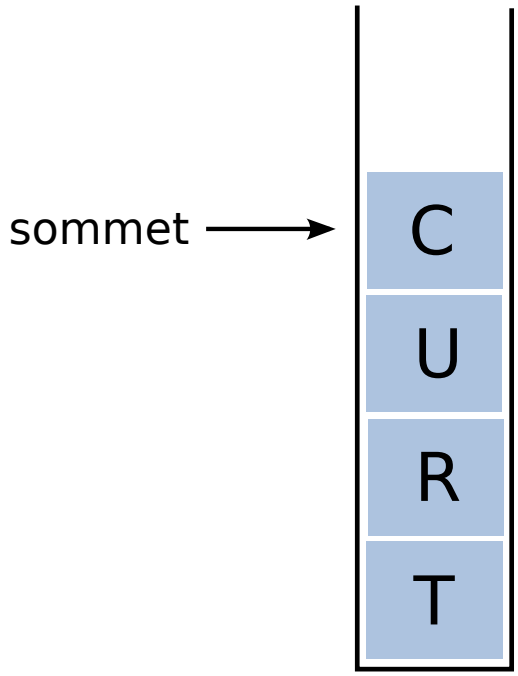
Inverser une chaîne de caractères



Exemple

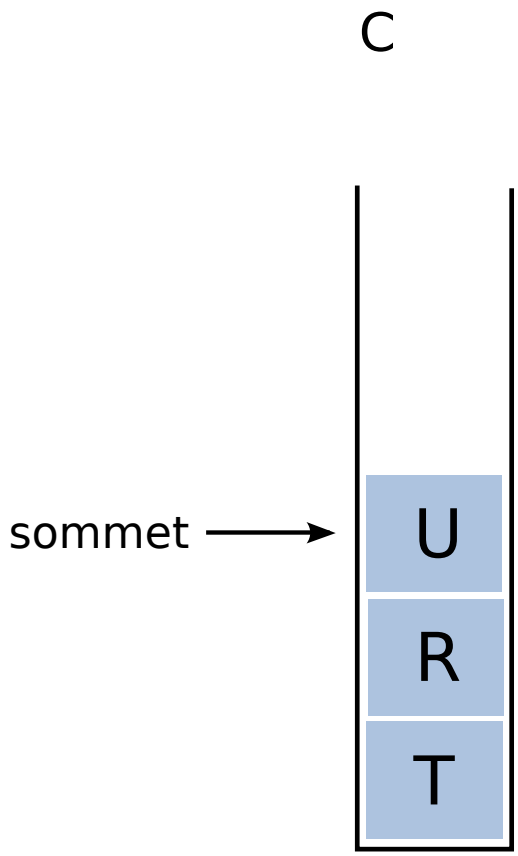
Inverser une chaîne de caractères

TRUC



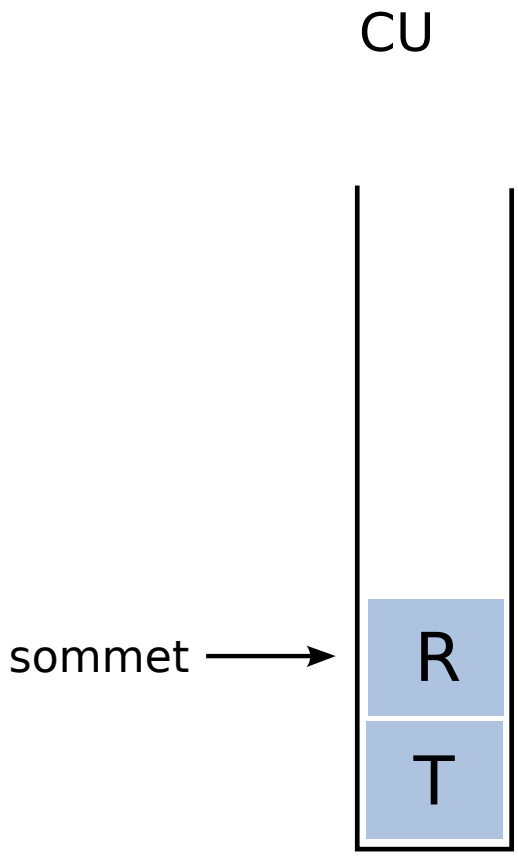
Exemple

Inverser une chaîne de caractères



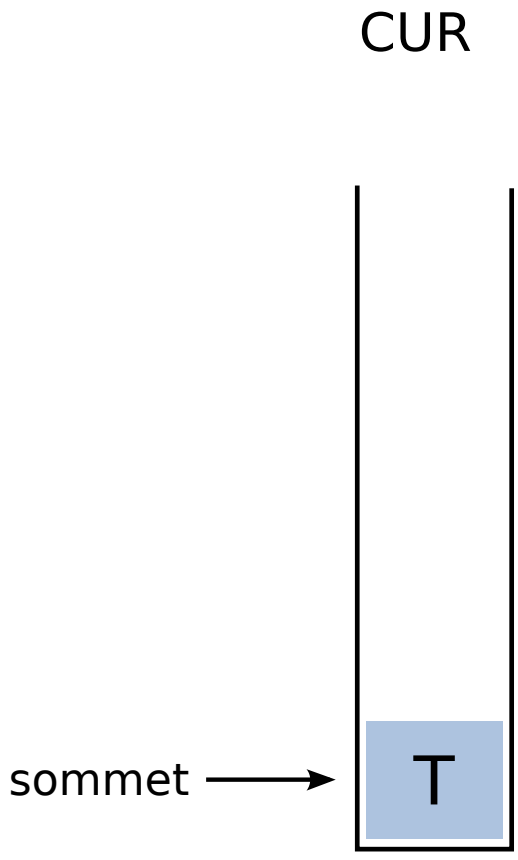
Exemple

Inverser une chaîne de caractères



Exemple

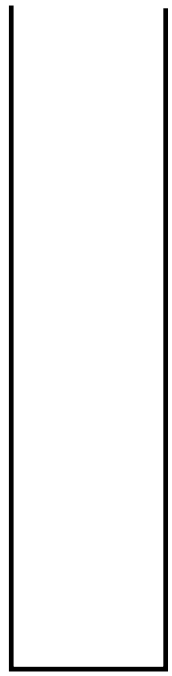
Inverser une chaîne de caractères



Exemple

Inverser une chaîne de caractères

CURT



Code Python

Inverser une chaîne de caractères

Des files, pourquoi ?

Quelques exemples d’utilisation :

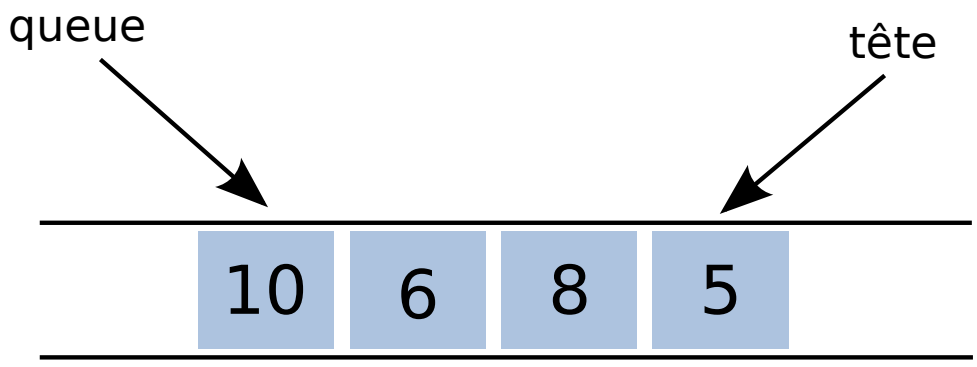
- File d’attente dans un magasin
- Gestion de stocks de denrées périssables
- Ordonnancement de processus

File

- Une file est un conteneur d’éléments qui respecte le principe du *First In First Out (FIFO) : le premier élément* accessible est le premier ajouté.
- Le premier élément enlevé est le plus ancien de la file

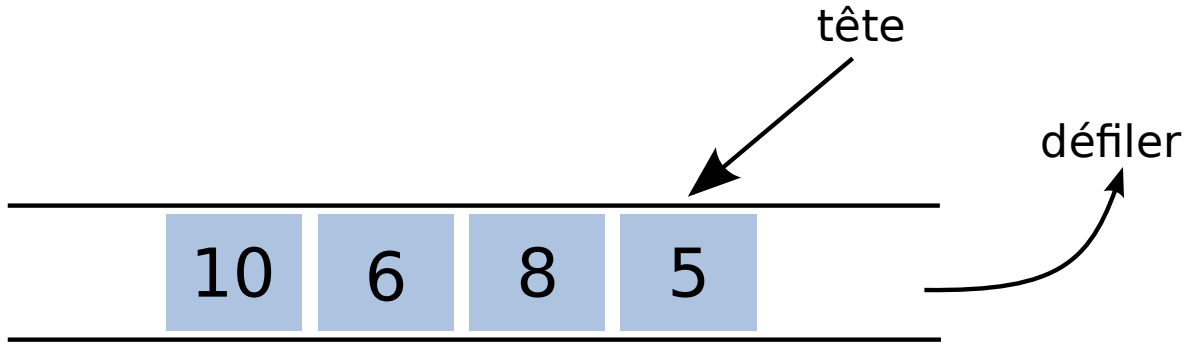
File : FIFO

File à l’origine :



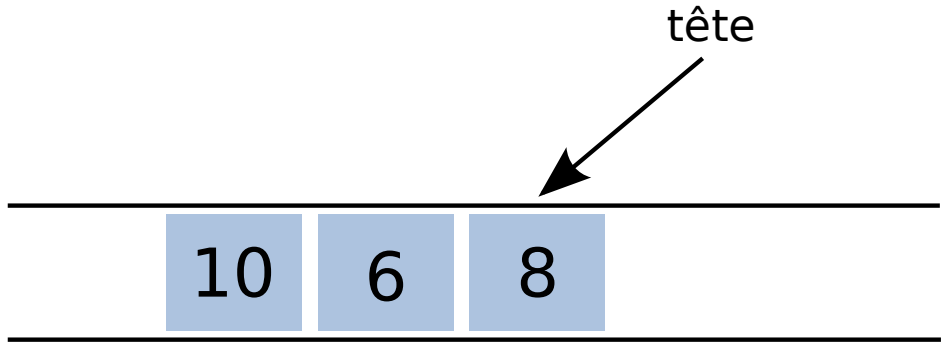
File : FIFO

On défile la valeur 5 en tête :



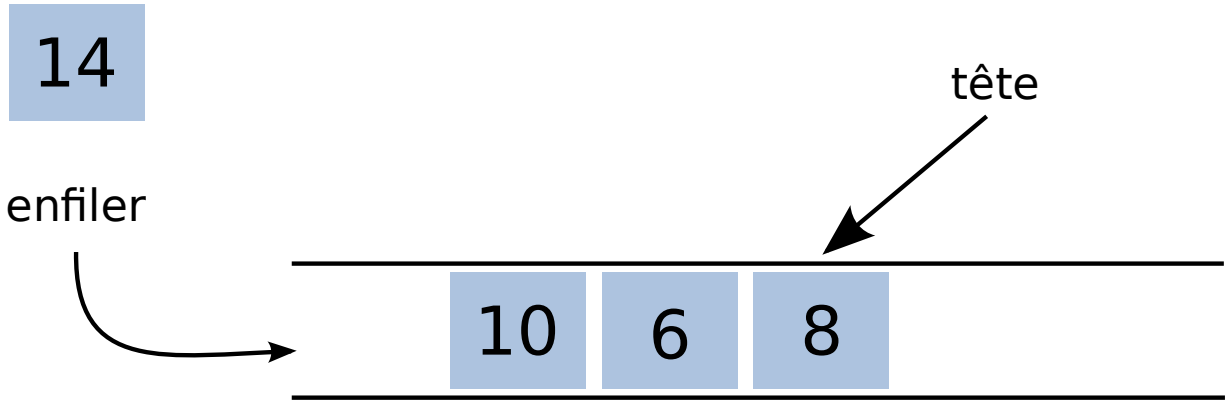
File : FIFO

8 devient la valeur en tête de ma file :



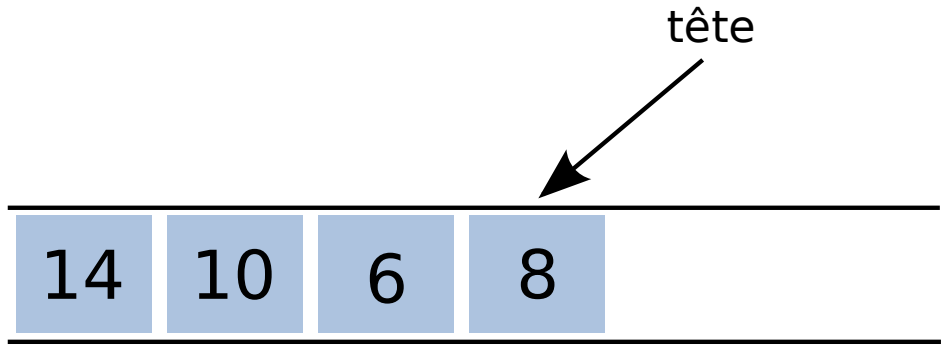
File : FIFO

On enfile 14 :



File : FIFO

14 devient la valeur en queue de file :

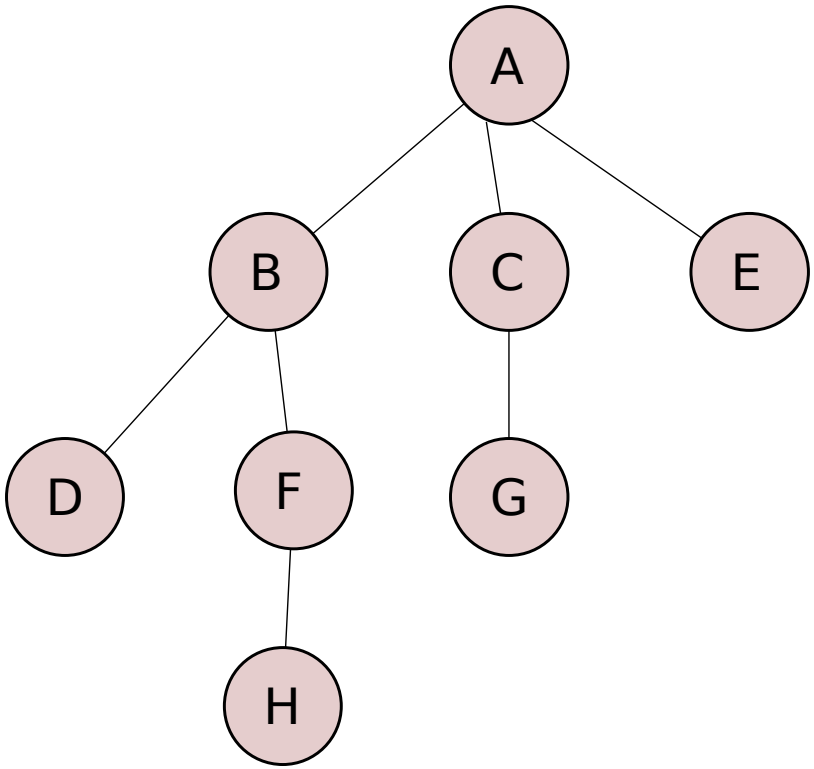


File

Opérations

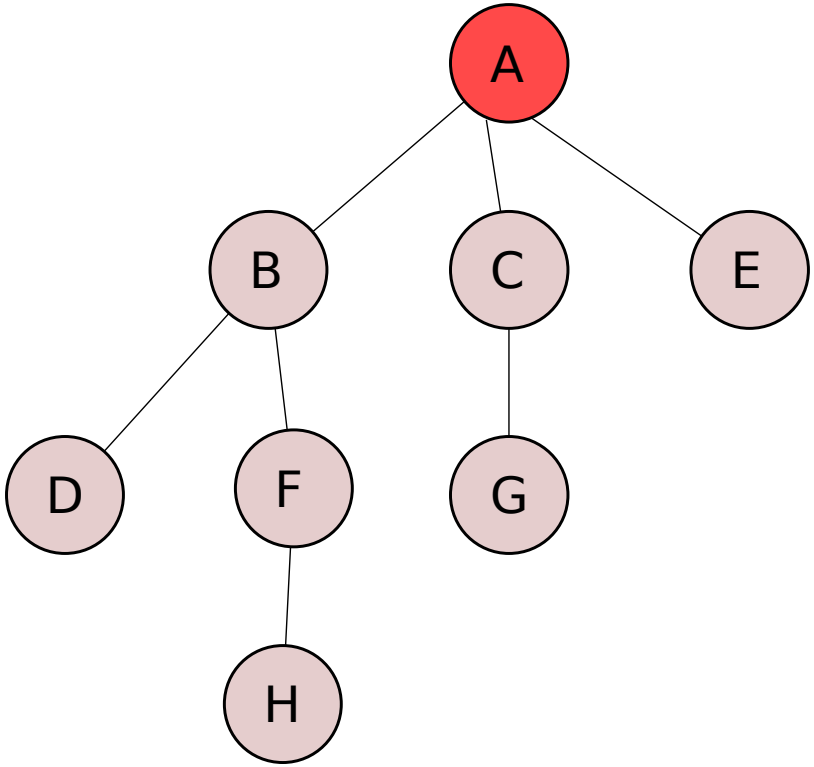
Le type de données abstrait File spécifie plusieurs opérations appelées primitives :

Application : Parcours en largeur

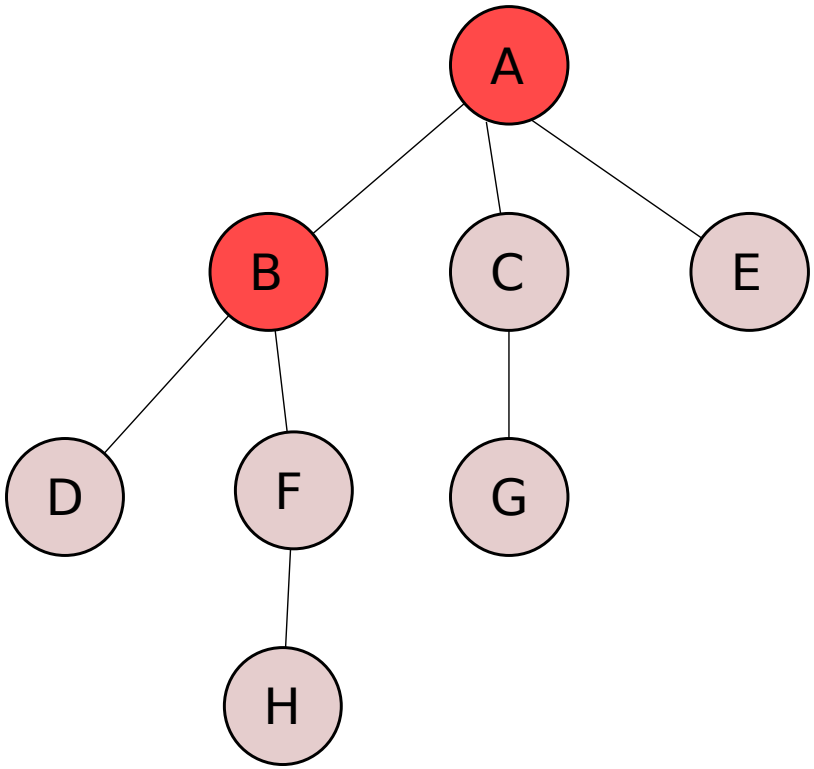


A

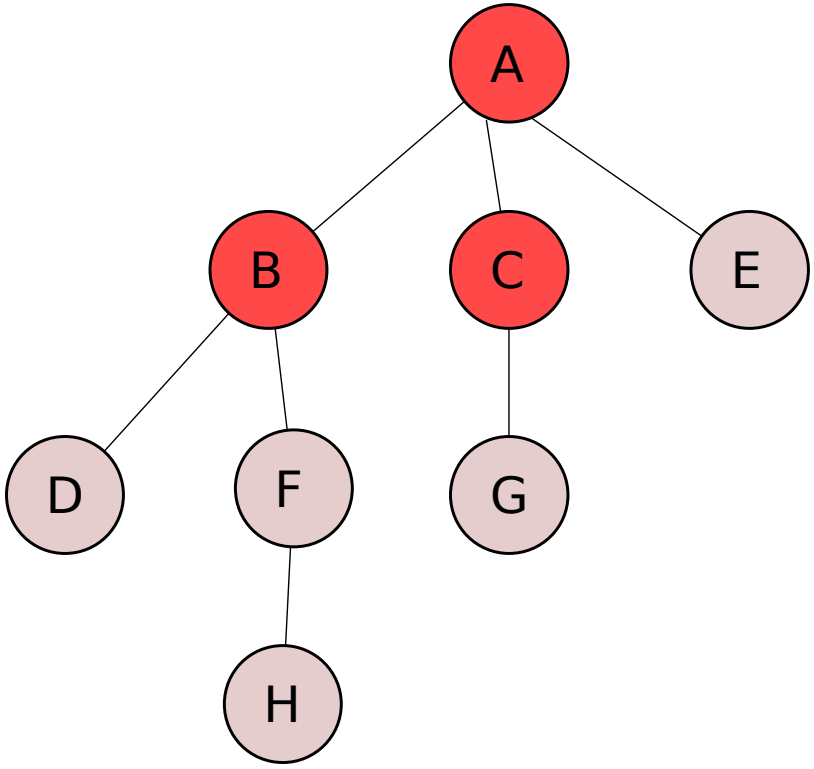
Application : Parcours en largeur



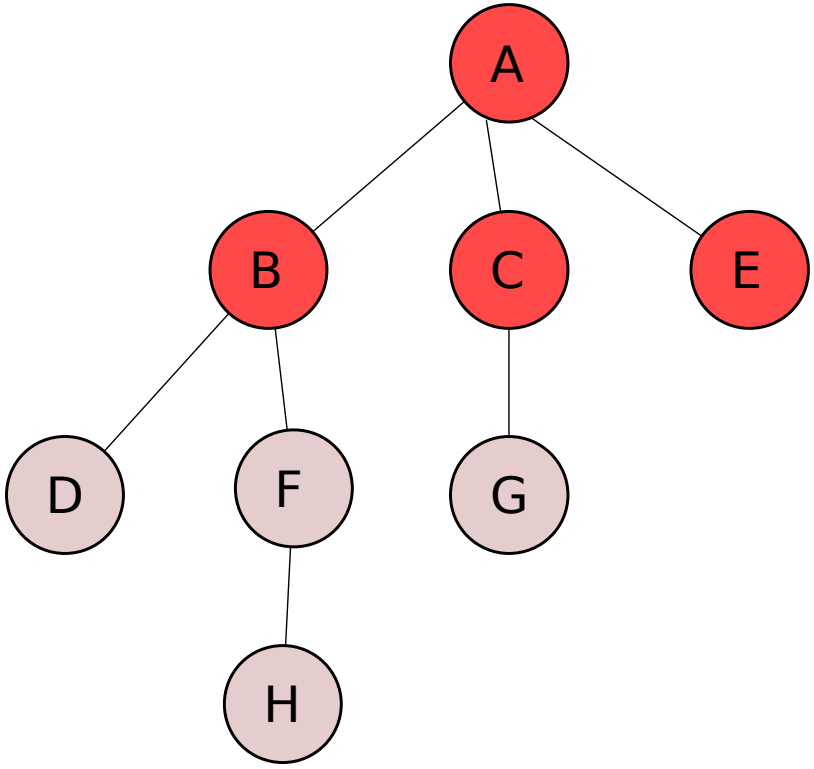
Application : Parcours en largeur



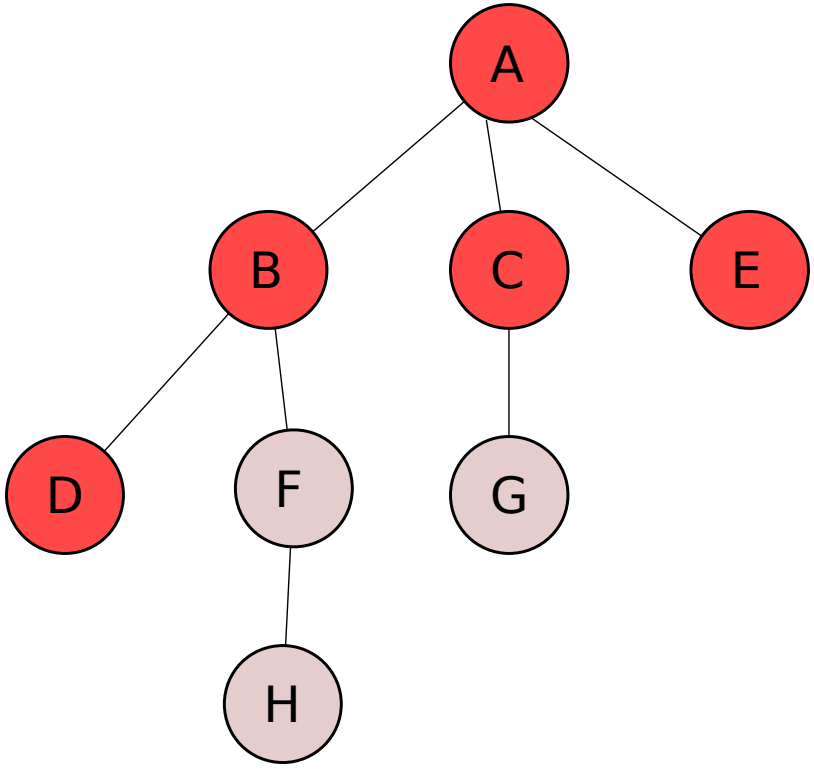
Application : Parcours en largeur



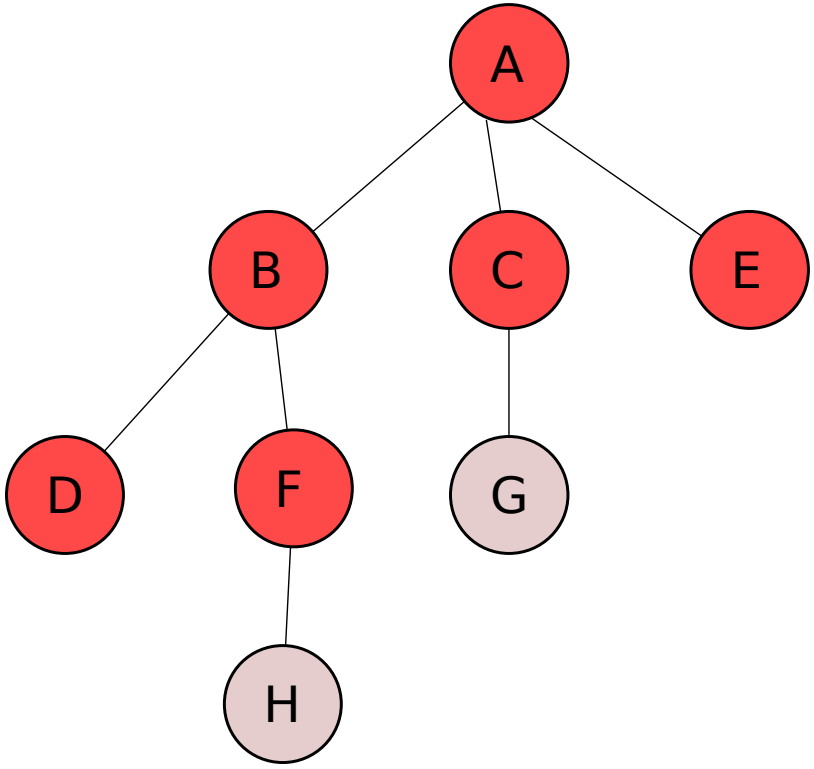
Application : Parcours en largeur



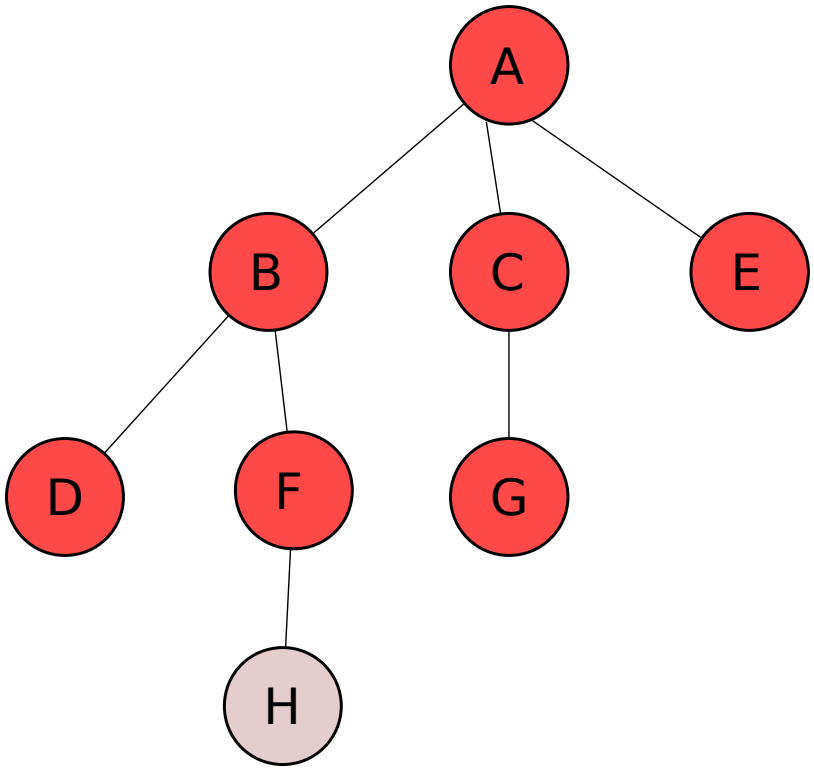
Application : Parcours en largeur



Application : Parcours en largeur

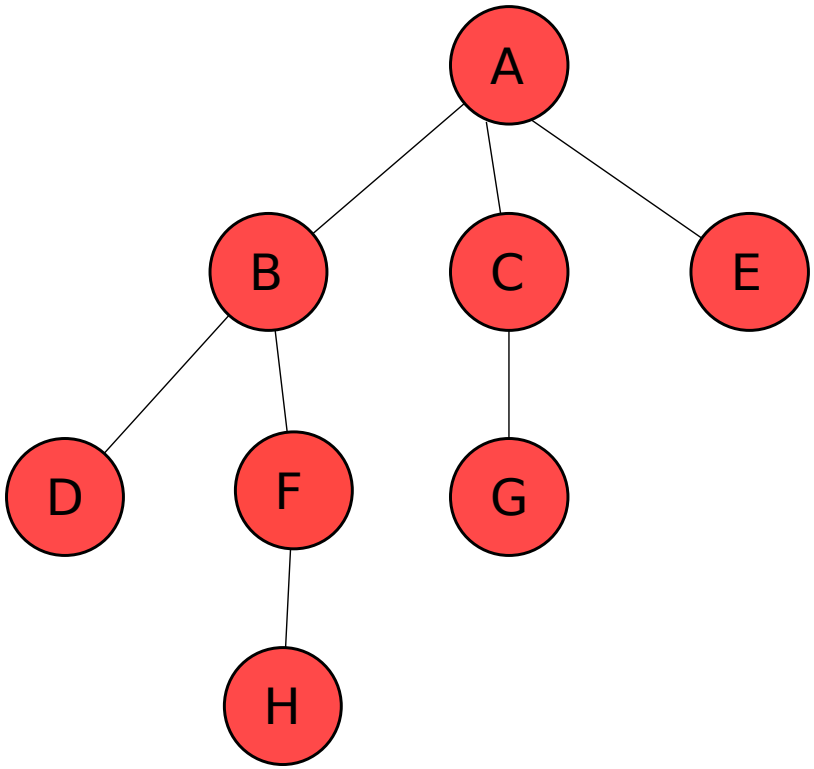


Application : Parcours en largeur



H

Application : Parcours en largeur



Ordre de visite : A-B-C-E-D-F-G-H

Bilan

Dans ce cours nous avons vu :

- Deux nouveaux types de données abstraits,
- Avec un accès limité à un seul élément
- Les piles : le dernier élément ajouté ⇒ premier sorti (LIFO)
- Les files : le premier élément ajouté ⇒ premier sorti (FIFO)

Bilan

Files et des piles

- insertion facile d'un élément
(au début pour une pile, à la fin pour une file)
- suppression facile de l'élément élément au début
Mais :
- accès “difficile” au n-ième élément

Tableaux

- accès facile au n-ième élément : tab[i-1]
Mais
- insertion “difficile” d'un élément
- suppression “difficile” d'un élément

Résumé

files V.S piles