

Algo 17 – Parcours en largeur d'abord d'un Arbre Binaire

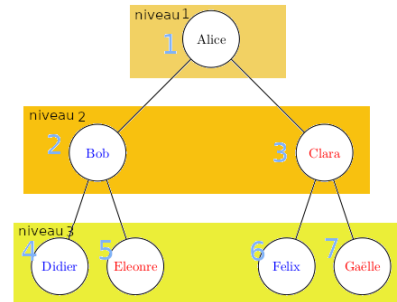


Fonction du parcours en programmation itératif :

```

1  def parcours_largeur(arbre:'AB') -> None:
2      """Exploration (et affichage) en largeur de l'arbre (version itérative)"""
3      if not est_ABV(arbre):
4          f = nv_FV()
5          enfiler(f, arbre)
6          while not est_file_vider(f):
7              a = defiler(f)
8              print(contenu(racine(a)))
9              g = gauche(a)
10             if not est_ABV(g):
11                 enfiler(f, g)
12             d = droite(a)
13             if not est_ABV(d):
14                 enfiler(f, d)

```



Fonction du parcours en programmation itératif :

```

1  def parcours_largeur_2(arbre:'AB') -> None:
2      """Exploration en largeur de l'arbre (version récursive)"""
3      if not est_ABV(arbre):
4          f = nv_FV()
5          enfiler(arbre, f)
6          exploration_suivante(f)
7
8  def exploration_suivante(f:'File') -> None:
9      """Affiche la prochaine racine et
10     stocke dans la File f les deux prochains sous-arbres détectés"""
11     if est_FV(f): # Cas de base
12         return None
13     else: # Cas récursif
14         a = defiler(f)
15         print(contenu(racine(a)))
16         g = gauche(a)
17         if not est_ABV(g):
18             enfiler(f, g)
19         d = droite(a)
20         if not est_ABV(d):
21             enfiler(f, d)
22         return exploration_suivante(f)

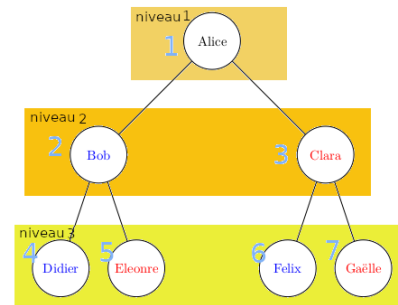
```

Fonction du parcours en programmation itératif :

```

1  def parcours_largeur(arbre:'AB') -> None:
2      """Exploration (et affichage) en largeur de l'arbre (version itérative)"""
3      if not est_ABV(arbre):
4          f = nv_FV()
5          enfiler(f, arbre)
6          while not est_file_vide(f):
7              a = defiler(f)
8              print(contenu(racine(a)))
9              g = gauche(a)
10             if not est_ABV(g):
11                 enfiler(f, g)
12             d = droite(a)
13             if not est_ABV(d):
14                 enfiler(f, d)

```



Fonction du parcours en programmation itératif :

```

1  def parcours_largeur_2(arbre:'AB') -> None:
2      """Exploration en largeur de l'arbre (version récursive)"""
3      if not est_ABV(arbre):
4          f = nv_FV()
5          enfiler(arbre, f)
6          exploration_suivante(f)
7
8  def exploration_suivante(f:'File') -> None:
9      """Affiche la prochaine racine et
10     stocke dans la File f les deux prochains sous-arbres détectés"""
11     if est_FV(f): # Cas de base
12         return None
13     else: # Cas récursif
14         a = defiler(f)
15         print(contenu(racine(a)))
16         g = gauche(a)
17         if not est_ABV(g):
18             enfiler(f, g)
19         d = droite(a)
20         if not est_ABV(d):
21             enfiler(f, d)
22         return exploration_suivante(f)

```

