MPSI

Sem. 2

Informatique

Cours

Ch. 6
Introduction aux graphes

E. Durif – X. Pessoles
J.-P. Berne
MPSI

1 Pile

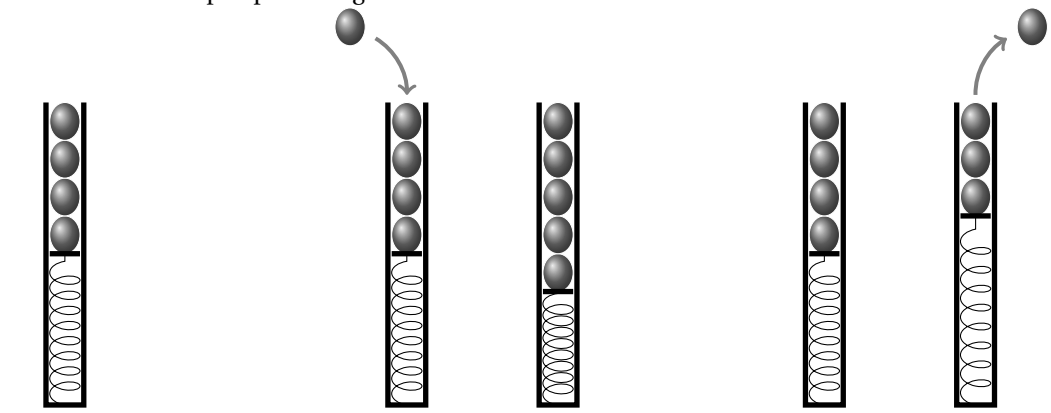
1.1 Présentation

Définition Pile Une pile est une structure de données dans laquelle le dernier élément stocké est le premier à en sortir. On parle de principe *LIFO* pour *Last In First Out*. Le dernier élément stocké est appelé **sommet**.

Pour gérer une pile, indépendamment de la façon dont elle est implémentée, on suppose exister les opérations élémentaires suivantes :

- `cree_pile()` qui créeune pile vide ;
- `empile(p,x)` qui empile l'élément `x` au sommet de la pile `p` ;
- `depile(p)` qui supprime le sommet de la pile `p` et renvoie sa valeur ;
- `est_vide(p)` qui teste si la pile `p`est vide.

On peut illustrer la structure de pile par l'image suivante :



pile (*stack*) empile (*push*) dépile (*pop*)

Théoriquement, chacune de ces opérations doit se faire à **temps constant** (complexité notée $\mathcal{O}(1)$).

Une des possiblités pour implémenter les piles est d'utiliser le module `deque`. Chacun des éléments de la pile peut être un objet de type différent.

```
from collections import deque

# Création d'une pile vide
pile = deque()

# Test si une pile est vide
len(pile) == 0

# Ajout de l'élément Truc au sommet de la pile
pile.append("Truc")

# Suppression (et renvoi) du sommet d'une pile non vide
sommet = pile.pop()
```

1.1.1 File

Définition File Une file est une structure de données dans laquelle le premier élément stocké est le premier à en sortir. On parle de principe *FIFO* pour *First In First Out*.

Pour gérer une file, indépendamment de la façon dont elle est implémentée, on suppose exister les opérations élémentaires suivantes :

- création d'une file vide ;
- test si une file est vide ;
- rajout d'un élément dans la file ;
- suppression (et renvoi) du premier élément innséré dans la file.

Théoriquement, chacune de ces opérations doit se faire à **temps constant**.

Une des possiblités pour implémenter les piles est d'utiliser le module `deque`. Chacun des éléments de la file peut être un objet de type différent. Dans cette vision des files, les éléments sont ajoutés « à droite » et sortent de la file « par la gauche ».

```
from collections import deque

# Création d'une file vide
```

```
file = deque()

# Teste si une pile est vide
len(file) == 0

# Ajoute l'élément Truc dans la file
file.append("Truc")

# Suppression (et renvoi) du premier élément inséré dans la file
sommet = pile.popleft()
```

1.2 Parcours générique d'un graphe

Définition Parcours générique Soit in graphe $G = (S, A)$. On parle de parcours générique d'un graphe lorsqu'on souhaite savoir, à partir du sommet S_i, quels sont les sommets accessibles.

1.3 Parcours en largeur

1.4 Parcours en profondeur

1.5 Détection de la présence des cycles

1.6 Connexité d'un graphe non orienté

2 Pondération d'un graphe

3 Recheche du plus court chemin

3.1 Algorithme de Dijkstra

3.2 Algorithme A⋆

Définition

Définition

Définition

Définition

Définition

S2