

Informatique en CPGE (2018-2019)

Les Piles

S. B.

Lycée des EK

12 septembre 2018

Pour manipuler des piles, trois fonctions sont absolument nécessaires.

- Une fonction "créer une pile" qui crée une pile vide et éventuellement lui alloue une certaine capacité.
- Une fonction "empiler" qui ajoute un élément sur la pile ; le terme anglais est "push". Dans le cas d'une pile à capacité finie, nous devons vérifier avant que la pile n'est pas pleine.
- Une fonction "dépiler" qui enlève le "sommet" de la pile et le renvoie ; le terme anglais est "pop". Nous devons ici vérifier que la pile n'est pas vide.

D'autres fonctions peuvent être utiles, en voici quelques-unes ici parmi les plus courantes.

- Une fonction "la pile est-elle vide ?" qui renvoie vrai si la pile est vide et faux sinon.
- Une fonction "taille de la pile" qui renvoie le nombre d'éléments stockés dans la pile.

- Une fonction "sommet de la pile" qui renvoie le sommet de la pile sans le dépiler ; le terme anglais est "peek".
- Une fonction "vider la pile" qui permet de dépiler tous les éléments ; le terme anglais est "clear".
- Des fonctions "dupliquer le sommet" et "échanger les deux premiers éléments" qui sont utilisées pour la notation polonaise inverse ; les termes anglais sont respectivement "dup" et "swap".

Nous allons simplement utiliser ici des objets connus avec certaines restrictions.

Nous créons une liste p (objet de type **list**), de longueur N suffisamment grand. Dans l'élément $p[0]$, nous stockons le nombre d'éléments de la pile.

```
def creer_pile(c) :      # crée une pile de capacité c
    p=(c+1)*[None]      # la pile est vide
    p[0]=0               # le nombre d'éléments de la pile
    return p
```

Attention à ne pas confondre la capacité qui est le nombre d'éléments que peut contenir une pile et la taille qui est le nombre d'éléments effectivement contenus dans une pile.

Nous définissons maintenant les fonctions "empiler" et "dépiler" :

```
def empiler(p,x) :      # ajout de l'élément x sur la pile
    assert p[0]<len(p)-1    # la pile n'est pas pleine ?
    p[0]=p[0]+1
    p[p[0]]=x

def depiler(p) :
    assert p[0]>0      # la pile n'est pas vide ?
    x=p[p[0]]
    p[p[0]]=None
    p[0]=p[0]-1
    return x
```

Voici les définitions de quelques fonctions utiles :

```
def pile_vide(p) :  
    return p[0]==0  
  
def taille(p) :  
    return p[0]  
  
def sommet(p) :  
    assert p[0]>0  
    return p[p[0]]
```

Remarque : il est important de bien comprendre que même si nous utilisons des listes sur lesquelles nous pouvons agir avec de nombreuses fonctions ou méthodes, une pile sera manipulée uniquement avec les fonctions spécifiques que nous créons.

Par exemple il est interdit d'écrire `pile[i]` pour obtenir un élément de la pile, même si cette instruction ne renvoie évidemment aucune erreur !

Nous utilisons ici les objets de type **list** avec les méthodes **append** et **pop** et la fonction **len** pour définir nos fonctions spécifiques aux piles.

```
def creer_pile() :  
    return []  
  
def empiler(p,x) :  
    p.append(x)  
  
def depiler(p) :  
    assert len(p)>0  
    return p.pop()
```

```
def pile_vide(p) :  
    return p==[ ]  
  
def taille(p) :  
    return len(p)  
  
def sommet(p) :  
    assert len(p)>0  
    return p[-1]
```

Dans un navigateur web, une pile sert à mémoriser les pages Web visitées. L'adresse de chaque nouvelle page visitée est empilée et l'utilisateur dépile l'adresse de la page précédente en cliquant sur un bouton "Afficher la page précédente".

La fonction "annuler la frappe" (ou "undo" en anglais) d'un logiciel de traitement de texte mémorise les modifications apportées au texte dans une pile et permet ainsi de revenir en arrière pas à pas.

Il est possible d'inverser les éléments contenus dans une liste, un tableau ou une chaîne de caractères en utilisant une pile. Il suffit d'empiler les éléments sur une pile puis de reconstituer la liste, le tableau ou la chaîne inverse en dépilant les éléments.

L'évaluation des expressions mathématiques en notation post-fixée, ou notation polonaise inverse, utilise une pile.

La notation polonaise inverse (NPI), permet d'écrire les formules arithmétiques sans utiliser de parenthèses ; pour cela les opérandes sont présentés avant les opérateurs.

Par exemple, l'expression " $3 + 4$ " s'écrit en NPI " $3\ 4\ +$ ", l'expression " $7 \times (2 + 3)$ " peut s'écrire sous la forme " $7\ 2\ 3\ +\ \times$ " ou " $2\ 3\ +\ 7\ \times$ ".

La réalisation d'une calculatrice NPI est basée sur l'utilisation d'une pile : les opérandes sont empilés un à un, et les résultats des calculs sont retournés en haut de la pile.

Pour l'expression " 3 + 4 " qui s'écrit en NPI " 3 4 + ", on empile les opérandes 3 puis 4 ; ensuite apparaît l'opérateur " + ", donc on dépile 4 puis 3, on les ajoute et on empile le résultat 7.

Pour l'expression " $7 \times (2 + 3)$ " qui peut s'écrire en NPI " $7\ 2\ 3\ +\ \times$ ", on empile 7 puis 2 puis 3;

l'opérateur " $+$ " apparaît donc on dépile 3 puis 2, on les ajoute et on empile le résultat 5.

La pile contient alors 7 et 5 ;

l'opérateur " \times " apparaît, donc on dépile 5 puis 7, on les multiplie et on empile le résultat 35. La pile contient alors l'élément 35.