Exercices pile - file correction

Christophe Viroulaud

Terminale - NSI

Archi 06

Sommaire

Exercice 1

```
1  def est_vide(p: list) -> bool:
2      return len(p) == 0
3
4  def empiler(p: list, e: int) -> None:
5      p.append(e)
6
7  def depiler(p: list) -> int:
8      return p.pop()
9
10 p = []
```

Code 1 – pile

```python
def est_vide(f: list) -> bool:
    return len(f) == 0

def enfiler(f: list, e: int) -> None:
    f.insert(0, e)

def defiler(f: list) -> int:
    return f.pop()

f = []
```

Code 2 – file

La modification de la taille d'un tableau a un coup qui peut être linéaire.

Sommaire

Exercice 2

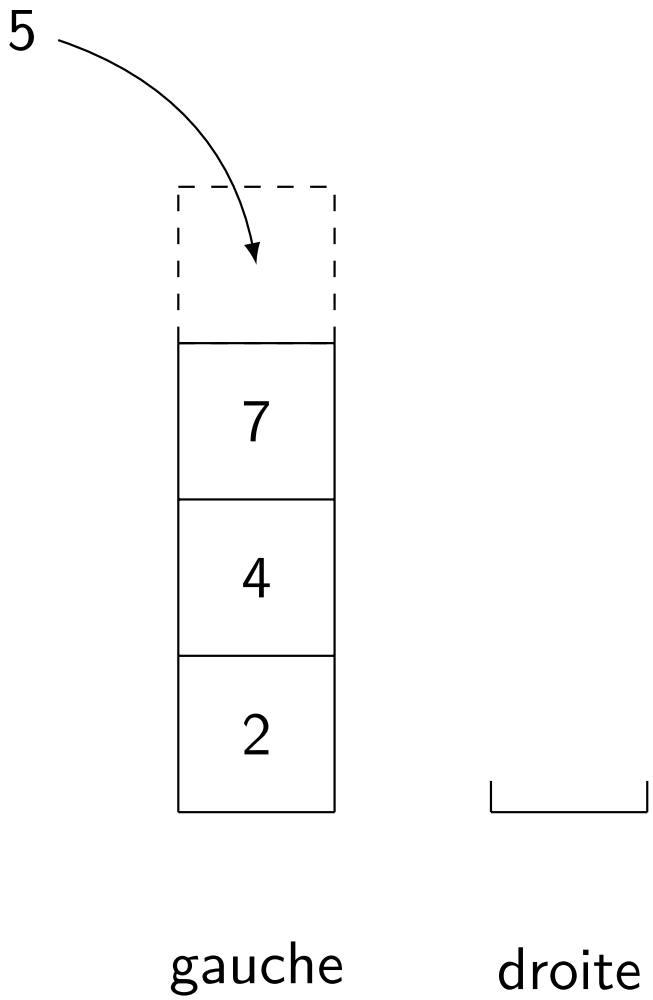

FIGURE 1 – enfiler

Le premier entré est 2.

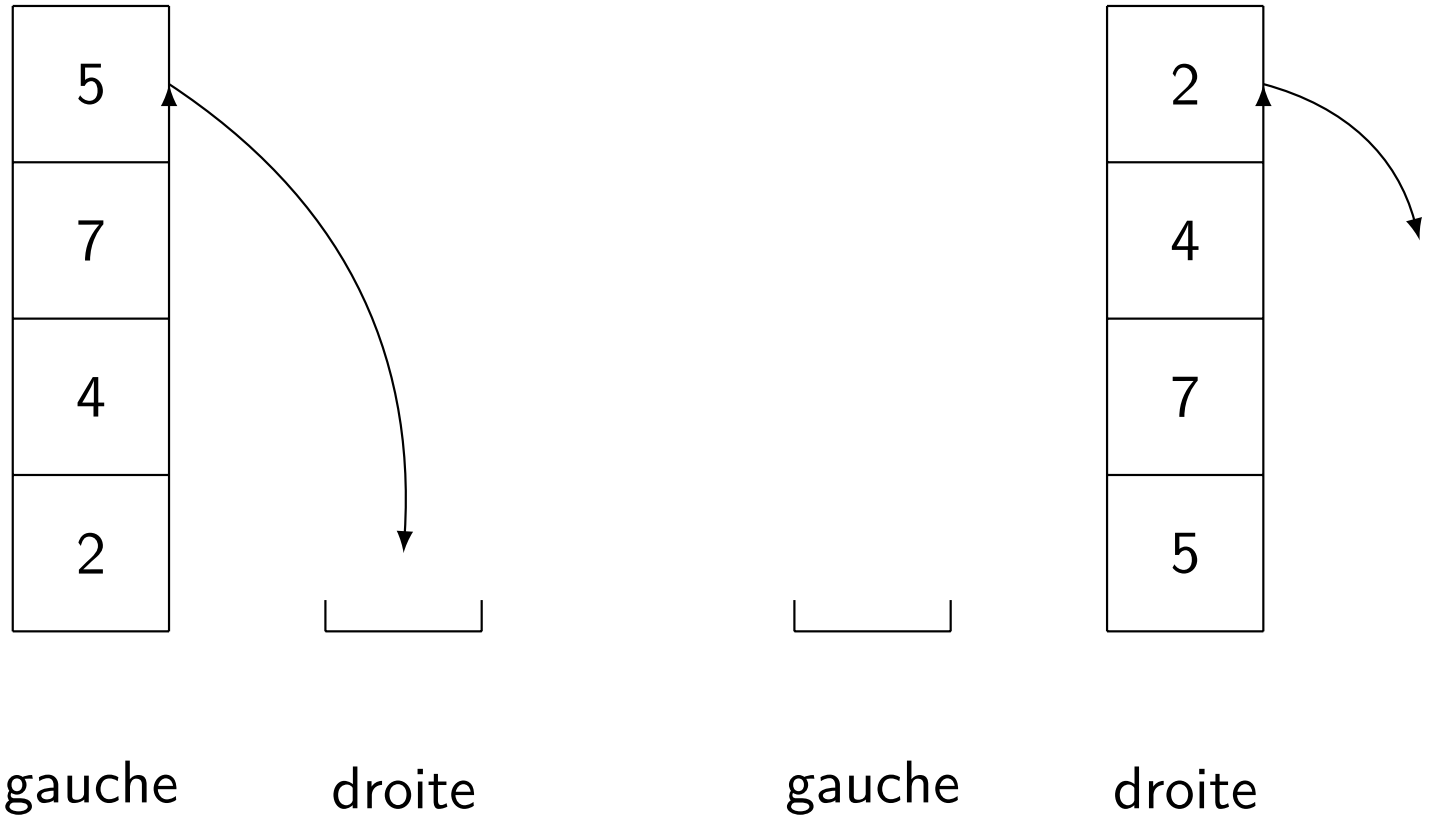

FIGURE 2 – défiler - cas 1

La pile droite est vide, on commence par dépiler celle de gauche.

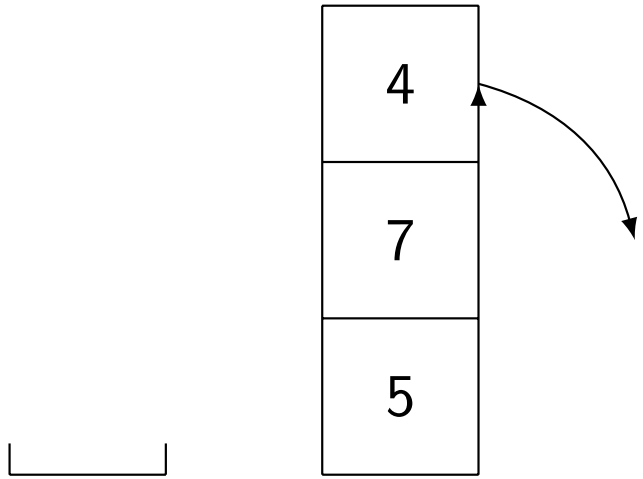

FIGURE 3 – défiler - cas 2

La pile droite n'est pas vide. On dépile normalement.

```
1  class File2:
2      def __init__(self):
3          self.gauche = Pile()
4          self.droite = Pile()
5
6      def est_vide(self) -> bool:
7          return self.gauche.est_vide() and
     self.droite.est_vide()
```

```
1  def enfiler(self, e: int) -> None:
2      self.gauche.empiler(e)
3
4  def defiler(self) -> int:
5      if self.droite.est_vide():
6          while not self.gauche.est_vide():
7              self.droite.empiler(self.gauche.
       depiler())
8
9      return self.droite.depiler()
```

Sommaire

Exercice 3

```
1  # Création du cercle
2  soldats = File()
3
4  for i in range(1, 42):
5      soldats.enfiler(i)
6
7  # Élimination tous les 3
8  while not(soldats.est_vide()):
9      # les non-éliminés reviennent dans la
       file
10     for _ in range(2):
11         soldats.enfiler(soldats.defiler())
12
13     # soldat éliminé
14     elimine = soldats.defiler()
15
16 # dernier éliminé
17 print(elimine)
```

Sommaire

Exercice 4

```
1  def bien_parenthesee(code: str) -> bool:
2      parentheses = Pile()
3      i = 0
4      correct = True
5      while i < len(code) and correct:
6          # empile une ouvrante
7          if code[i] == "(":
8              parentheses.empiler("(")
9          # dépile une ouvrante quand on trouve
       une fermante
10         elif code[i] == ")":
11             # si rien à dépiler -> mauvais
       parenthésage
12             if parentheses.depiler() is None:
13                 correct = False
14         i += 1
15
16     return correct
```

```
1  def bien_parenthesee_rec(code: str, i: int, p: Pile)
       -> bool:
2      if i == len(code):
3          return True
4      else:
5          if code[i] == "(":
6              p.empiler("(")
7          elif code[i] == ")":
8              if p.depiler() is None:
9                  return False
0          return bien_parenthesee_rec(code, i+1, p)
```

Sommaire

Exercice 6

```
1  def polonaise(chaine: str) -> int:
2      p = Pile()
3      for e in chaine.split():
4          if e == "+" or e == "*":
5              val1 = p.depiler()
6              val2 = p.depiler()
7              if e == "+":
8                  p.empiler(val1+val2)
9              else:
0                  p.empiler(val1*val2)
1          else:
2              p.empiler(int(e))
3
4      return p.depiler()
```