

Bibliothèque de exercice commenté

Christophe Viroulaud

Terminale - NSI

Archi 07

En utilisant la pile construite à partir d'un tableau, mesurer la durée d'exécution pour :

- ▶ empiler 100000 éléments,
- ▶ dépiler 100000 éléments.

```
1 class Pile:
2     def __init__(self):
3         self.donnees = []
4
5     def est_vide(self) -> bool:
6         return self.donnees == []
7
8     def empiler(self, e: int) -> None:
9         self.donnees.append(e)
10
11    def depiler(self) -> int:
12        if not self.est_vide():
13            return self.donnees.pop()
```

```
1 p = Pile()
2 deb = time()
3 for i in range(NB):
4     p.empiler(i)
5 fin = time()
6 print("empiler ", fin-deb)
7
8 deb = time()
9 for i in range(NB):
10    p.depiler()
11 fin = time()
12 print("dépiler ", fin-deb)
```

```
1 empiler  0.04126310348510742
2 dépiler  0.06682586669921875
```

Effectuer les mêmes mesures pour la file :

```
1 class File:
2     def __init__(self):
3         self.donnees = []
4
5     def est_vide(self) -> bool:
6         return self.donnees == []
7
8     def enfiler(self, e: int) -> None:
9         self.donnees.append(e)
10
11    def defiler(self) -> int:
12        if not self.donnees == []:
13            return self.donnees.pop(0)
```

```
1 f = File()
2 deb = time()
3 for i in range(NB):
4     f.enfiler(i)
5 fin = time()
6 print("enfiler ", fin-deb)
7
8 deb = time()
9 for i in range(NB):
10    f.defiler()
11 fin = time()
12 print("défiler ", fin-deb)
```

```
1 enfiler  0.041974782943725586
2 défiler  2.0046255588531494
```

À retenir

Python propose des outils optimisés dans la bibliothèque `collections` :

<https://docs.python.org/fr/3/library/collections.html>

Écrire une nouvelle classe `File` en utilisant une `deque`.

```
1 class File2:
2     def __init__(self):
3         self.donnees = deque()
4
5     def est_vide(self) -> bool:
6         return len(self.donnees) == 0
7
8     def enfiler(self, e: int) -> None:
9         self.donnees.append(e)
10
11    def defiler(self) -> int:
12        if not self.est_vide():
13            return self.donnees.popleft()
```

Observation

L'**interface** reste identique pour l'utilisateur de la classe.


```
1 f = File2()
2 deb = time()
3 for i in range(NB):
4     f.enfiler(i)
5 fin = time()
6 print("enfiler ", fin-deb)
7
8 deb = time()
9 for i in range(NB):
10    f.defiler()
11 fin = time()
12 print("défiler ", fin-deb)
```

```
1 enfiler  0.02210521697998047
2 défiler  0.038460731506347656
```