Semaine 11

Initiation à l'algorithmique et programmation

Revekka Kyriakoglou

Qu’est-ce qu’un tableau ?

- Un tableau est une structure de données utilisée pour stocker des éléments du même type.
- C’est une collection d’éléments stockés à des emplacements mémoire contigus.
- L’idée est de stocker plusieurs éléments du même type ensemble, ce qui facilite le calcul de la position de chaque élément en ajoutant simplement un décalage à une valeur de base.
- Les tableaux sont utilisés pour implémenter d’autres structures de données, telles que les listes, les piles, les files d’attente et leurs dérivés.

Propriétés des tableaux

- **Taille fixe :** Une fois définie, la taille d'un tableau ne peut pas être changée.
- **Efficacité :** Les tableaux permettent un accès efficace à leurs éléments grâce à l'indexation.
- **Disposition en mémoire :** Les éléments d'un tableau sont stockés dans des emplacements mémoire contigus, ce qui optimise la performance des opérations.
- **Accès direct :** Chaque élément d'un tableau peut être directement accédé en utilisant son numéro d'index.

Différence avec la liste

- **Homogénéité de type :** Les tableaux stockent des éléments du même type alors que les listes peuvent stocker des éléments de types différents.
- **Taille :** Les tableaux ont une taille fixe. Les listes sont dynamiques et peuvent croître au besoin.
- **Performance :** Les éléments du tableau sont stockés de manière contiguë en mémoire, ce qui conduit à une meilleure performance du cache. Les listes, étant dynamiques, peuvent entraîner un surcoût de mémoire.
- **Fonctionnalité :** Les listes viennent avec une variété de méthodes intégrées que les tableaux ne supportent pas, telles que l'ajout, l'extension et l'insertion.

Pourquoi Utiliser des Tableaux ? (1/2)

- **Efficacité en Termes d'Utilisation de la Mémoire :** Les tableaux allouent de la mémoire en blocs contigus. Cela minimise le surcoût de mémoire et peut conduire à des améliorations significatives de performance, surtout pour de grands ensembles de données.
- **Accès Rapide :** La capacité d'indexation directe des tableaux offre un accès rapide à tout élément. Ceci est particulièrement utile dans des scénarios où des opérations critiques en temps sont nécessaires, comme dans les systèmes temps réel.
- **Simplicité pour les Données Linéaires :** Pour des données qui forment naturellement une séquence (par exemple, des séries temporelles, des simulations numériques), les tableaux offrent une manière simple de stocker et de manipuler l'information.

Pourquoi Utiliser des Tableaux ? (2/2)

- **Fondation pour des Structures Plus Complexes :** De nombreuses structures de données et algorithmes avancés commencent avec des tableaux comme mécanisme de stockage sous-jacent. Comprendre les tableaux est crucial pour saisir ces concepts plus complexes.
- **Optimisation Matérielle et Langage :** De nombreux langages de programmation et architectures matérielles sont optimisés pour l'accès séquentiel à la mémoire, rendant les tableaux un choix efficace pour une large gamme de tâches informatiques.
- **Applicabilité dans le Calcul Mathématique et Scientifique :** Les tableaux sont largement utilisés dans des domaines comme la physique, l'ingénierie et l'informatique pour les calculs mathématiques, la modélisation et les simulations en raison de leurs caractéristiques de structure et de performance.

Syntaxe et Codes de Type dans les Tableaux

- **Syntaxe d'un Tableau :** Pour créer un tableau en Python, vous utilisez la fonction array du module array, qui nécessite un code de type et une liste d'initialisation.
  `array(typecode, [initialiser])`

```
from array import array
a = array('i', [1, 2, 3, 4])
```

- **Comprendre les Codes de Type :** Le code de type spécifie le type des éléments du tableau. Chaque code représente un type de données différent.
  - 'i' - entier signé
  - 'f' - point flottant
  - 'd' - double précision de point flottant

Codes de Type dans les Tableaux

 Choisir le bon code de type est essentiel pour une utilisation efficace de la mémoire et pour garantir l'intégrité des données.

`https://docs.python.org/fr/3/library/array.html`

Codes de Type dans les Tableaux

 Choisir le bon code de type est essentiel pour une utilisation efficace de la mémoire et pour garantir l'intégrité des données.

`https://docs.python.org/fr/3/library/array.html`

 Visualisation de l'exécution par l'utilisation du site :

`http://pythontutor.com`

Accéder aux éléments d'un tableau

- **Accès direct :** Utilisez l'index entre crochets pour accéder à un élément. Rappelez-vous, l'indexation commence à 0.

```
print(a[0])  # Sortie : 1
```

- **Tranchage :** Accédez à une plage d'éléments en utilisant la syntaxe de tranchage.

```
print(a[1:3])  # Sortie : array('i', [2, 3])
```

- **Itération :** Bouclez sur le tableau en utilisant une boucle for pour accéder à chaque élément.

```
for element in a:
    print(element)
```

Méthodes de tableau

- **append() :** Ajoute un nouvel élément à la fin du tableau.

```
a.append(5)  # Ajoute 5 à la fin
```

- **extend() :** Ajoute des éléments d'un autre tableau ou itérable à la fin.

```
a.extend([6, 7])  # Étend le tableau avec 6, 7
```

- **pop() :** Supprime le dernier élément et le retourne.

```
a.pop()  # Supprime et retourne le dernier élément
```

- **insert() :** Insère un élément à une position spécifiée.

```
a.insert(0, 0)  # Insère 0 au début
```

- **remove() :** Supprime la première occurrence d'un élément.

```
a.remove(2)  # Supprime la première occurrence de 2
```

Gestion des tableaux en Python : ‘array‘ vs ‘numpy‘ (1/2)

- **Utilisation de ‘from array import array‘ :**
  - Le module ‘array‘ fait partie de la bibliothèque standard de Python. Il fournit un objet tableau plus efficace qu’une liste pour certains cas d’utilisation.
  - Syntaxe :
    `from array import array`
  - Cas d’usage :
    Lorsque vous avez besoin d’un tableau simple et typé pour le stockage de base et des opérations élémentaires sans nécessiter des méthodes mathématiques avancées.

Gestion des tableaux en Python : ‘array‘ vs ‘numpy‘(2/2)

- **Utilisation de ‘numpy‘ :**
  - ‘numpy‘ est une bibliothèque tierce offrant un puissant objet tableau, adapté au calcul numérique. Les tableaux ‘numpy‘ sont idéaux pour les calculs scientifiques et numériques car ils prennent en charge une large gamme d’opérations mathématiques.
  - Syntaxe :
    `import numpy as np`
  - Cas d’usage :
    Lorsque vous travaillez avec de grands ensembles de données, avez besoin de performances élevées pour les calculs numériques ou requérez des opérations mathématiques avancées et des fonctionnalités telles que la diffusion, la vectorisation, l’algèbre linéaire, etc.