Les algorithmes de tri

Qu'est ce qu'un algorithme de tri?

Quels sont les principaux algorithmes de tri?

Qu’est ce qu’un algorithme de tri?

Un algorithme de tri est une méthode ou un ensemble de règles que l'on suit pour réorganiser les éléments d'une collection (comme un tableau ou une liste) dans un certain ordre, généralement croissant ou décroissant.

Les differents types des algorithmes de tri:

01

TRI À BULLES

02

TRI PAR FUSION

03

TRI RAPIDE

04

TRI PAR INSERTION

05

TRI PAR SELECTION

Tri à bulles

Compare et échange les éléments adjacents jusqu'à ce que la liste soit triée.

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
```

15/11/2024

Tri par fusion

Divise la liste en moitiés, trie chaque moitié, puis fusionne les moitiés triées

```python
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        left_half = arr[:mid]
        right_half = arr[mid:]

        merge_sort(left_half)
        merge_sort(right_half)

        i = j = k = 0

        while i < len(left_half) and j < len(right_half):
            if left_half[i] < right_half[j]:
                arr[k] = left_half[i]
                i += 1
            else:
                arr[k] = right_half[j]
                j += 1
            k += 1

        while i < len(left_half):
            arr[k] = left_half[i]
            i += 1
            k += 1

        while j < len(right_half):
            arr[k] = right_half[j]
            j += 1
            k += 1
    return arr
```

15/11/2024

Tri rapide

Utilise un pivot pour diviser et trier les sous-listes.

```
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quick_sort(left) + middle + quick_sort(right)
```

Tri par insertion

Insère chaque élément à sa place dans une liste triée

```python
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key
    return arr
```

15/11/2024

Tri par sélection

Sélectionne l'élément minimum et le place au début de la liste, répète jusqu'à ce que la liste soit triée.

```python
def selection_sort(arr):
    for i in range(len(arr)):
        min_idx = i
        for j in range(i+1, len(arr)):
            if arr[j] < arr[min_idx]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
    return arr
```

Importance des Algorithmes de Tri

- Les algorithmes de tri sont cruciaux car ils:
- Améliorent l'efficacité des opérations de recherche.
- Facilitent l'analyse et la manipulation des données.
- Sont utilisés comme étapes intermédiaires dans d'autres algorithmes plus complexes.