DSA · FICHE PRATIQUE

Cheatsheet : Recherches et tris

Choisir selon ordre initial, stabilité, mémoire et garanties.

Par Ali ZAINOUL · Cours gratuit

Définitions, contrats et coûts

Choisir selon ordre initial, stabilité, mémoire et garanties.

Recherches et tris

Notion	À retenir
Recherche linéaire	O(n) au pire ; pas de précondition de tri.
Recherche binaire	O(log n) comparaisons avec accès indicé et données triées.
Insertion / sélection	O(n²) au pire ; insertion est utile sur petits préfixes presque triés.
Fusion / rapide	Fusion : Θ(n log n), mémoire auxiliaire à préciser ; rapide : O(n log n) moyen, O(n²) au pire selon pivot et partition.

Vérifier sur un exemple

Trouver une borne d'insertion

bisect_left cherche la première position où une valeur pourrait être insérée dans une séquence triée.

Python

```python
# binary_search.py
from bisect import bisect_left
values = [1, 3, 3, 8]
index = bisect_left(values, 3)
found = index < len(values) and values[index] == 3
print(index, found)
# Expected output:
# 1 True
```

Résultat expliqué. La vérification finale distingue une position d'insertion d'une valeur trouvée. La recherche est logarithmique ; l'insertion physique dans une liste reste Θ(n).

Erreur à éviter. Une recherche binaire sur des données non triées ne respecte pas son contrat.

▼ **Un tri stable conserve-t-il toute la position initiale des éléments ?**

Il conserve l'ordre relatif des éléments dont les clés de comparaison sont égales.

Télécharger cette fiche en PDF · Toutes les fiches DSA

Version web : https://www.clearcode.fr/courses/programming/dsa/cheatsheet/search-sort