STRUCTURES DE DONNÉES ET ALGORITHMES · CHAPITRE 8

Tester une appartenance et éliminer les doublons

Distinguez ensemble et séquence. Comparez déduplication, appartenance, union et intersection sans supposer un ordre de résultat commun.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Distinguer appartenance, multiplicité, union et intersection.
- Un ensemble conserve l'appartenance, mais pas la multiplicité.
- L'union réunit les éléments ; l'intersection conserve ceux présents des deux côtés.

Prérequis. Avoir étudié les chapitres DSA précédents ; connaître collections, parcours et coûts asymptotiques.

Partir d'un besoin

Pour savoir si un identifiant a déjà été rencontré, faut-il conserver ses répétitions ? Un ensemble conserve l'appartenance, mais pas la multiplicité de chaque valeur.

Définir la structure et les conventions

Un **ensemble** contient chaque valeur au plus une fois. L'**appartenance** demande si une valeur est présente. L'**union** réunit les valeurs présentes dans au moins un ensemble ; l'**intersection** conserve celles présentes dans les deux. L'ordre des éléments ne fait pas partie du contrat mathématique de l'ensemble.

Python construit un set avec set(values) ; le test `value in seen` demande l'appartenance. Pour un ensemble vide, écrivez set(), car {} crée un dictionnaire. Les éléments doivent être hachables : leur hachage doit être compatible avec leur égalité et rester stable pendant leur utilisation. Une

liste Python modifiable ne convient donc pas comme élément de set.

MATLAB utilise ici unique sur un vecteur numérique pour retirer les répétitions et ismember pour tester l'appartenance. Cette méthode représente les valeurs distinctes par un vecteur ; elle n'est pas une traduction de la structure interne du set Python. sorted(seen) sert seulement à donner un affichage Python prévisible.

Suivre les états

Valeurs distinctes de [3, 1, 3, 2]

Demande	Résultat
Nombre de valeurs distinctes	3
3 appartient-il à l'ensemble ?	Vrai
4 appartient-il à l'ensemble ?	Faux
Valeurs affichées en ordre croissant	[1, 2, 3]

Comparer les réalisations

Les valeurs distinctes sont 1, 2 et 3 ; 3 est présent. Le tri de l'affichage Python est une étape supplémentaire. Ne déduisez pas d'un affichage particulier que tous les ensembles possèdent un ordre croissant.

Pseudo-code

```
// example.pseudo
// distinct_values
// Keep each value once: [3, 1, 3, 2] -> {1, 2, 3}.
// The braces here denote a mathematical set, not an ordered sequence.
// Membership of 3 is TRUE; multiplicity is not retained.
```

Python

```python
# distinct_values.py
values = [3, 1, 3, 2]
seen = set(values)
present = 3 in seen
ordered = sorted(seen)
print(ordered, present)
```

MATLAB

```matlab
% distinct_values.m
values = [3, 1, 3, 2];
seen = unique(values);
present = ismember(3, seen);
ordered = seen;
disp(ordered);
disp(present);
```

Justifier et compter

Un set fondé sur le hachage permet l'appartenance O(1) en moyenne sous les mêmes hypothèses que le dictionnaire ; le pire cas peut être O(n). Trier k valeurs distinctes pour l'affichage ajoute O(k log k). unique et ismember de MATLAB ne reçoivent pas ici une garantie de temps constant : leur stratégie et les représentations sont différentes.

Prévoir puis expérimenter

Prévoyez le résultat pour une entrée vide et pour [2, 2, 2]. Comparez ensuite A = {(1, 2)} et B = {(2, 3)} : union {(1, 2, 3)}, intersection {2}. En Python, | réalise l'union des sets et & leur intersection.

Exercice corrigé

▼ Un ensemble suffit-il pour compter les ventes de chaque produit ?

Non : les occurrences sont perdues. Il indique quels produits ont été vus ; un dictionnaire de compteurs conserve leur nombre de ventes.

À retenir

- Un ensemble conserve l'appartenance, mais pas la multiplicité.
- L'union réunit les éléments ; l'intersection conserve ceux présents des deux côtés.
- Un tri d'affichage ajoute son propre coût et ne fait pas partie du contrat mathématique d'ensemble.

Version web : https://www.clearcode.fr/courses/programming/dsa/associative/sets