STRUCTURES DE DONNÉES ET ALGORITHMES · CHAPITRE 10

Éliminer un sous-arbre à chaque comparaison

Appliquez l'invariant d'un arbre binaire de recherche, suivez une recherche et reliez son coût à la hauteur plutôt qu'au seul nombre de nœuds.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Utiliser l'ordre des sous-arbres pour rechercher une clé.
- L'ordre d'un arbre de recherche porte sur tous les descendants, pas seulement sur les enfants immédiats.
- Une recherche suit un seul enfant après chaque comparaison et coûte O(h + 1).

Prérequis. Avoir étudié les chapitres DSA précédents ; connaître liens, piles et appels récursifs.

Partir d'un besoin

Pour chercher 10 dans un arbre de racine 8, peut-on ignorer tout le côté gauche ? Seulement si une règle relie les valeurs de chaque sous-arbre à celle de son parent.

Définir la structure et les conventions

Dans un **arbre binaire de recherche** à clés distinctes, toutes les clés du sous-arbre gauche sont strictement inférieures à celle du nœud ; toutes celles du sous-arbre droit sont strictement supérieures. Cette propriété s'applique à tous les descendants, pas seulement aux enfants immédiats. Nous excluons les doublons ; une autre politique devrait être définie explicitement.

Une recherche compare la cible à la valeur courante. En cas d’égalité, elle réussit. Sinon, l’invariant autorise à suivre un seul enfant. Rencontrer la sentinelle signifie que la valeur est absente. Les tableaux values, left et right représentent les mêmes trois nœuds qu’au chapitre précédent.

Insérer une nouvelle clé suit les mêmes comparaisons jusqu’au lien absent, puis y rattache une nouvelle feuille. Supprimer est plus délicat : une feuille se détache ; un nœud à un enfant peut être remplacé par cet enfant ; un nœud à deux enfants peut être remplacé par son successeur, la plus petite clé de son sous-arbre droit, qu’il faut ensuite retirer à son ancien emplacement.

Suivre les états

Recherche de 10

Nœud	Comparaison	Suite
8	10 > 8	Enfant droit
10	10 == 10	Trouvé

Comparer les réalisations

Le résultat est vrai pour 10, faux pour 7. Les enfants absents ne sont jamais lus. En MATLAB, enregistrez search_tree.m avec sa fonction locale à la fin du script. Le nom contains_key évite de confondre cette opération avec une recherche de texte.

Pseudo-code

```
// example.pseudo
// search_tree
WHILE node != -1 DO
    IF target == values[node] THEN
        RETURN TRUE
    ELSE IF target < values[node] THEN
        node <- left[node]
    ELSE
        node <- right[node]
    END IF
END WHILE
RETURN FALSE
```

Python

```
# search_tree.py
values = [8, 3, 10]
left = [1, -1, -1]
right = [2, -1, -1]
def contains(target):
    node = 0
    while node != -1:
        if target == values[node]:
            return True
        if target < values[node]:
            node = left[node]
        else:
            node = right[node]
    return False
found = contains(10)
print(found)
```

MATLAB

```
% search_tree.m
values = [8, 3, 10];
left = [2, 0, 0];
right = [3, 0, 0];
found = contains_key(10, values, left, right);
disp(found);
function found = contains_key(target, values, left, right)
    node = 1;
    found = false;
    while node ~= 0
        if target == values(node)
            found = true;
            return;
        end
        if target < values(node)
            node = left(node);
        else
            node = right(node);
        end
    end
end
```

Justifier et compter

Une recherche descend au plus h + 1 nœuds : O(h + 1) temps et Θ(1) mémoire auxiliaire dans cette version itérative. Un arbre équilibré maintient h = O(log n) ; un arbre dégénéré en chaîne peut avoir h = n - 1. Les arbres AVL ou rouge-noir ajoutent des règles et des rééquilibrages pour garantir une hauteur logarithmique ; notre représentation ne les réalise pas.

Prévoir puis expérimenter

Dessinez les insertions successives 1, 2, 3, 4 dans un arbre vide sans rééquilibrage. Toutes les clés se placent à droite. Comparez la recherche de 4 avec celle dans un arbre dont la racine vaut 2.

Exercice corrigé

▼ Vérifier seulement que chaque enfant gauche est inférieur à son parent suffit-il ?

Non. Une valeur 12 sous le côté gauche d'une racine 8 peut respecter son parent immédiat tout en violant l'invariant global. Il faut contrôler les bornes héritées de tous les ancêtres.

À retenir

- L'ordre d'un arbre de recherche porte sur tous les descendants, pas seulement sur les enfants immédiats.
- Une recherche suit un seul enfant après chaque comparaison et coûte O(h + 1).
- Sans rééquilibrage, la hauteur peut atteindre n - 1 et la recherche devenir linéaire.

Version web : https://www.clearcode.fr/courses/programming/dsa/hierarchies/binary-search-trees