STRUCTURES DE DONNÉES ET ALGORITHMES · CHAPITRE 11

Faire remonter la plus petite priorité

Distinguez tas et arbre de recherche, calculez les indices parent-enfants et insérez une priorité en restaurant l'invariant du tas minimum.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Restaurer l'invariant d'un tas minimum après une insertion.
- Le tas minimum place un minimum à la racine, sans trier tout le tableau.
- La remontée d'une insertion et la descente d'un retrait restaurent l'ordre parent-enfant.

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

Partir d'un besoin

Une file classique traite la première tâche arrivée. Si une tâche plus urgente doit passer d'abord, il faut une file de priorité. Un tas permet de consulter rapidement sa plus petite clé.

Définir la structure et les conventions

Un **tas minimum binaire** est un arbre binaire complet : tous les niveaux sauf éventuellement le dernier sont remplis, et le dernier se remplit de gauche à droite. Chaque parent possède une clé inférieure ou égale à celles de ses enfants. La racine contient donc un minimum. Les frères ne sont pas nécessairement triés.

Un tableau stocke les nœuds niveau par niveau. Avec des indices à partir de 0, les enfants de i sont 2i + 1 et 2i + 2, s'ils existent ; le parent d'un nœud non racine est floor((i - 1)/2). À partir de 1, les enfants sont 2i et 2i + 1, et le parent floor(i/2).

Pour insérer, on ajoute la nouvelle valeur en fin du tableau, puis on l'échange avec son parent tant qu'elle est plus petite. Python fournit heappush dans heapq. MATLAB montre explicitement cette remontée. Pour retirer le minimum, on remplace la racine par la dernière valeur, puis on la fait descendre en échangeant avec le plus petit enfant jusqu'à restaurer l'invariant.

Suivre les états

Insérer 1 dans [2, 5, 7]

Étape	Tas stocké en tableau
Ajouter en fin	[2, 5, 7, 1]
Échanger avec 5	[2, 1, 7, 5]
Échanger avec 2	[1, 2, 7, 5]

Comparer les réalisations

Le tas devient [1, 2, 7, 5] et son minimum vaut 1. Le fait que 7 précède 5 dans le tableau ne viole pas l'invariant : ces valeurs ne sont pas dans une relation parent-enfant qui exigerait l'ordre inverse.

Pseudo-code

```
// example.pseudo
// heap_insert
// The new value is already appended; indices start at 0.
WHILE index > 0 DO
    parent <- floor_divide(index - 1, 2)
    IF heap[parent] <= heap[index] THEN
        BREAK
    END IF
    SWAP heap[parent] AND heap[index]
    index <- parent
END WHILE
```

Python

```
# heap_insert.py
from heapq import heappush
heap = [2, 5, 7]
heappush(heap, 1)
minimum = heap[0]
print(heap, minimum)
```

MATLAB

```
% heap_insert.m
heap = [2, 5, 7, 1];
index = numel(heap);
while index > 1
    parent = floor(index / 2);
    if heap(parent) <= heap(index)
        break;
    end
    temporary = heap(parent);
    heap(parent) = heap(index);
    heap(index) = temporary;
    index = parent;
end
minimum = heap(1);
disp(heap);
disp(minimum);
```

Justifier et compter

Le tas complet possède O(log n) niveaux. Consulter le minimum est Θ(1) ; restaurer le tas après insertion ou retrait effectue O(log n) comparaisons. La croissance du tableau sous-jacent a son propre coût : un ajout ponctuel peut réallouer ; MATLAB reçoit déjà la case ajoutée dans cet exemple. Construire un tas par descentes depuis les derniers parents se fait en Θ(n), tandis que n insertions séparées coûtent O(n log n).

Prévoir puis expérimenter

Insérez plutôt 9 : aucun échange n’est nécessaire après son ajout. Puis vérifiez chaque paire parent-enfant de [1, 2, 7, 5] ; ne demandez pas que le tableau soit entièrement trié.

Exercice corrigé

▼ Un tas permet-il de rechercher une clé quelconque en O(log n) comme un arbre de recherche équilibré ?

Non. Son invariant ne sépare pas les petites clés à gauche et les grandes à droite. Une recherche de valeur arbitraire peut visiter Θ(n) nœuds.

À retenir

- Le tas minimum place un minimum à la racine, sans trier tout le tableau.
- La remontée d'une insertion et la descente d'un retrait restaurent l'ordre parent-enfant.
- Le tas favorise l'extraction de priorité ; il ne donne pas une recherche logarithmique de clé quelconque.

Version web : https://www.clearcode.fr/courses/programming/dsa/hierarchies/heaps