Data Structures & Algorithms

Topic: Heap

By

Ravi Kant Sahu

Asst. Professor,

Lovely Professional University, Punjab

Contents

- Introduction
- Insertion in Heap
- Deleting the Root of Heap
- Heap Sort
- Heap Sort Complexity
- Review Questions

Introduction

- **Heap (MaxHeap):** A complete binary tree H with n elements is called a Heap if each node N of H has the following property:

  " The value at N is greater than or equal to the value at each of the children of N."

- If the value at N is less than or equal to the value at each of the children of N, then it is called MinHeap.

- Heaps are maintained in memory by using linear array TREE.

Insertion in a Heap

- INSERT_HEAP(TREE, N, ITEM)

1. Set N = N+1, and PTR = N.
2. Repeat Step 3 to 6 while PTR > 1
3. Set PAR = $\lfloor PTR / 2 \rfloor$
4. If ITEM <= TREE [PAR], then:
   Set TREE[PTR] = ITEM and Return.
5. Else: Set TREE[PTR] = TREE[PAR].
6. Set PTR = PAR.
7. Set TREE[1] = ITEM.
8. Return.

Example

- Create a Heap from the following list of numbers:

  40, 30, 50, 20, 60, 55, 70, 60, 65, 50

Deletion of Root of a Heap

- Assign Root R to some variable ITEM.

- Replace the deleted node R by the last node L of Heap H so that H is still complete tree, but not a Heap.

- Call MaxHeapify to maintain the heap property.

Build MaxHeap

BUILD_MAX-HEAP(A)

1. heapsize[A] = length[A]
2. Repeat for i = ⌊ length[A]/2 ⌋ to 1
3. Call MAX_HEAPIFY(A, i)
4. Exit

Maintaining Heap Property

MAX_HEAPIFY(A, i)

1. Set: l = LEFT (i)
2. Set: r = RIGHT (i)
3. If l <= heapsize [A] and A[l] > A[i], then:
4. largest = l.
5. Else: largest = i.
6. If r <= heapsize [A] and A[r] > A[largest], then:
7. largest = r.
8. If largest != i, then:
9. Exchange A[i] $\longleftrightarrow$ A[largest].
10. MAX_HEAPIFY (A, largest).

Heap Sort

HEAP_SORT (A)

1. BUILD_MAXHEAP (A)
2. Repeat for i = length[A] to 2
3. Exchange A[1] $\leftrightarrow$ A[i].
4. Heapsize[A] = Heapsize [A] $-$ 1.
5. Call MAX_HEAPIFY(A, 1).
6. Exit.

Complexity of Heap Sort

Average and Worst case Complexity of Heap sort = O(nlogn).

Questions

Review Questions

- What do you mean by Heap?
- What is the complexity of Heap sort?
- Calculate the complexity of Heap sort.
- How will you insert an element in a heap?
- What is the difference between Maxheap and Minheap?