Heap data structure

Data structure The heap is a special case of a balanced binary tree data structure, where the root node's key is compared to its children and arranged accordingly.

What is the Heap data structure?

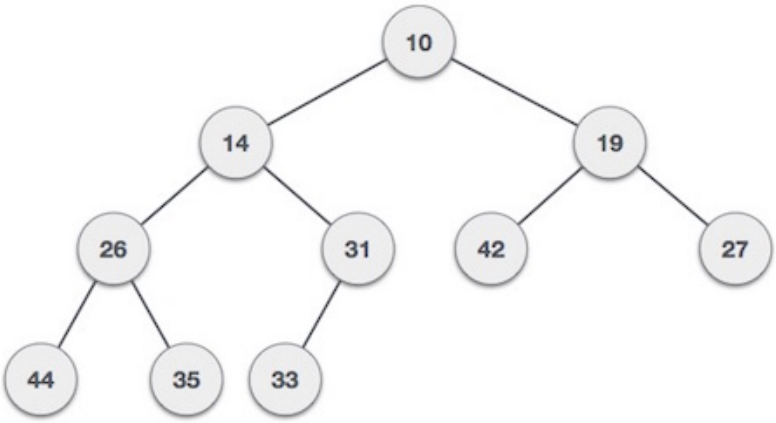
Data structure The heap is a special case of a balanced binary tree data structure, where the root node's key is compared to its children and arranged accordingly. If ? has a child node ? then:

key (?) ? key (?)

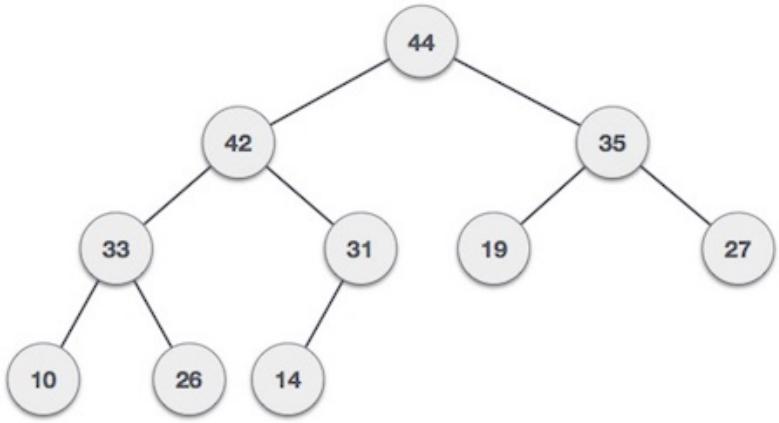
When the value of the parent node is greater than the value of the child node, this attribute creates a Max Heap. Based on this criterion, a Heap can be one of two types:

```
With data filled in ? 35 33 42 10 14 19 27 44 26 31
```

Min-Heap : here the value of the root node is less than or equal to the values ??of the child nodes.



Max-Heap : here the value of the root node is greater than or equal to the value of the child nodes.



The above two example trees are all built on the same input data and the same order.

Max Heap construction algorithm

We will use the same example to illustrate how to create a Max Heap. The method to build Min Heap is similar.

We will deduce an algorithm for Max Heap by inserting an element at a time. At any time, Heap must maintain (obey) its properties. During the insert process, we also assume that we are inserting a node in HEAPIFIED Tree.

B??c 1 : T?o m?t nút m?i t?i v? trí cu?i cùng c?a Heap. **B??c 2** : Gán giá tr? m?i cho nút này. **B??c 3** : So sánh giá tr? c?a nút con v?i giá tr? cha. **B??c 4** : N?u giá tr? c?a cha là nh? h?n con thì tráo ??i chúng. **B??c 5** : L?p l?i b??c 3 và 4 cho t?i khi v?n duy trì thu?c tính c?a Heap.

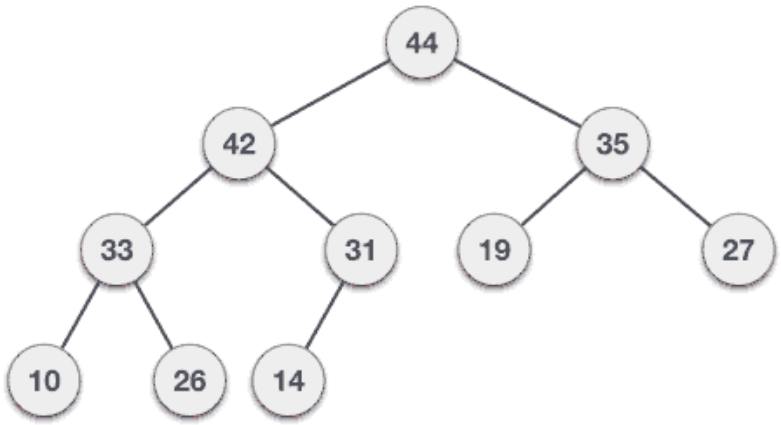
Note : In the Min Heap build algorithm, the value of the parent node will be smaller than the value of the child nodes.

For more details on Max Heap construction algorithm, let's look at the animated illustration below.

The algorithm deletes in Max Heap

Delete operation in Max (or Min) Heap always takes place at the root node and to delete the Largest (or Smallest) value. You follow the algorithm and animation below to understand more about this algorithm.

B??c 1 : Xóa nút g?c. **B??c 2** : Di chuy?n ph?n t? cu?i cùng có b?c th?p nh?t lên nút g?c. **B??c 3** : So sánh giá tr? c?a nút con này v?i giá tr? c?a cha. **B??c 4** : N?u giá tr? c?a cha là nh? h?n c?a con thì **tráo ??i** chúng. **B??c 5** : L?p l?i b??c 3 và 4 cho t?i khi v?n duy trì thu?c tính c?a Heap.



According to Tutorialspoint

Previous article: Spanning Tree in data structure and algorithm

Next lesson: Basics of recursion