Unit - 4 Advanced Data Structures

Heap

A heap is a binary tree with the following properties:

```
1. The value of each node is greater than or equal to the value of its
parent, with the maximum-value element at the root.
2. The left and right subtrees of each node are again heaps.
```

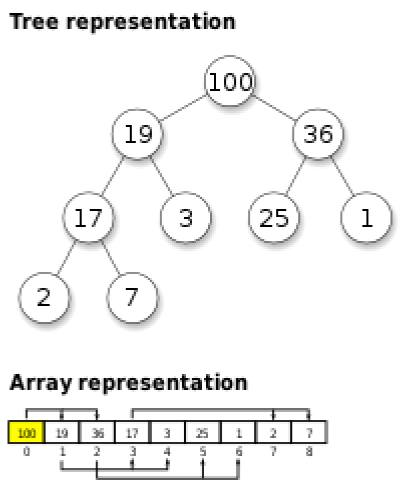
A heap tree is also a `Complete Binary Tree`.

There are two types of `Heaps`:

```
1. Max Heap
2. Min Heap
```

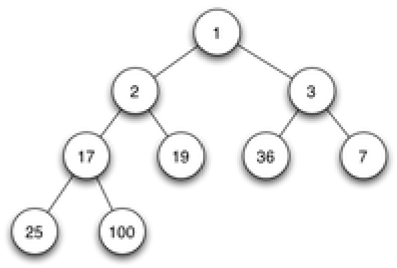
Max Heap: A `Max Heap` is when the parent node is greater than or equal to the value of its child nodes, and the root node is the largest value in the heap.

Example:



Min Heap: A `Min Heap` is when the parent node is less than or equal to the value of its child nodes, and the root node is the smallest value in the heap.

Example:



Heap Operations

```
1. Insertion
2. Deletion
3. Heapify
4. Build Heap
5. Heap Sort
```

Heap Tree Construction

Given keys are: 14, 24, 12, 11, 25, 8, 35 Now, constructing a max heap tree from the given keys.

```
    14
   /
  24
```

Now, as we see the property of max heap is not being followed we swap the values of 14 and 24.

```
    24
   /  \
  14    12
 /  \
```

11 25 Now, as we see the property is not being followed we swap the values of 14 and 25.

```
     24
   /    \
  25      12
 /  \    /  \
```

11 14 8 35 Now, as we see teh property is not being followed we swap the values of 12 and 35.

```
         24
       /    \
      25        35
     /  \      /  \
    11   14   8     12
```

Now, as we see the property is not being followed so we swap the values of 24 and 35.

```
            35
          /     \
      25        24
     /  \      /  \
    11   14   8     12
```

Now, as we see the property is being followed so we stop here, and the final `Max Heap` tree is created.

Question: Consider a Binary Max Heap implemented using an array. Which one of the following array represents a Binary Max Heap?

1. 25, 12, 16, 13, 10, 8, 14
2. 25, 14, 13, 16, 10, 8, 12
3. 25, 14, 16, 13, 10, 8, 12
4. 25, 14, 12, 13, 10, 8, 16

Answer: For the first array the tree is:

```
     25
    /  \
  12     16
 /  \   / \
13
```

Here as we see the property of max heap is not being followed so it is not a `Max Heap`.

Now, for the second array the tree is:

```
     25
    /  \
  14     13
 /  \   / \
```

16 10 8 12 Here as we see the property of max heap is not being followed so it is not a `Max Heap`.

Now, for the third array the tree is:

```
     25
    /  \
  14    16
 /  \   / \
```

13 10 8 12 Here as we see the property of max heap is being followed so it is a `Max Heap`.

Now, for the fourth array the tree is:

```
     25
    /  \
  14    12
 /  \   / \
```

13 10 8 16 Here as we see the property of max heap is not being followed so it is not a `Max Heap`.

Question: Whay is the content fo array after two delete operations on the following heap?

1. 14, 13, 12, 10, 8
2. 14, 12, 13, 8, 10
3. 14, 13, 8, 12, 10
4. 14, 13, 12, 8, 10

Answer: For the first array the tree is:

```
     14
    /  \
  13    12
 /  \
```

10 8 After the first delete operation the tree is:

```
     13
    /  \
  10    12
 /
```

8 After the second delete operation the tree is:

```
      12
     /
   10
   /
  8
```

So, the array is: 12, 10, 8

For the second array the tree is:

```
      14
     /  \
   12    13
  /  \
```

8 10 After the first delete operation the tree is:

```
       13
      /  \
    12    10
   /
  8
```

After the second delete operation the tree is:

```
       12
      /
    10
   /
  8
```

So, the array is: 12, 10, 8

For the third array the tree is:

```
      14
     /  \
   13    8
  /  \
```

12 10 After the first delete operation the tree is:

```
       13
      /  \
    12    8
```

```
  /
 10
```

After the second delete operation the tree is:

```
      12
     /
   10
  /
 8
```

So, the array is: 12, 10, 8

For the fourth array the tree is:

```
     14
    /  \
  13    12
 /  \
```

8 10 After the first delete operation the tree is:

```
      13
     /  \
   10    12
  /
 8
```

After the second delete operation the tree is:

```
      12
     /
   10
  /
 8
```

So, the array is: 12, 10, 8