1. Heap Properties

A heap is a special tree-based data structure that follows specific properties. The first property is the shape property, which states that a heap must be a complete binary tree. This means all levels are fully filled except possibly the last level, which is filled from left to right. The second property is the heap order property. In a max heap, the value of each parent node is greater than or equal to the values of its children. In a min heap, the value of each parent node is less than or equal to the values of its children. These properties ensure efficient access to the maximum or minimum element.

2. Heap Implementation Using Array

A heap can be efficiently implemented using an array by storing elements in a top-down and left-to-right manner. The root element is stored at the first position of the array. For any element stored at index i, its left child is stored at index 2i and its right child is stored at index 2i + 1. This method avoids using pointers and saves memory. Array representation makes insertion and deletion operations faster and simpler because the complete binary tree structure is naturally maintained.

3. Difference Between Binary Search Property and Heap Property

The binary search property ensures that all elements in the left subtree are smaller than the root and all elements in the right subtree are larger than the root. This property helps in efficient searching. The heap property, on the other hand, only ensures that parent nodes are either greater or smaller than their children depending on the heap type. Heap property does not maintain sorted order. Therefore, binary search trees are used for searching, while heaps are mainly used for priority-based operations.

4. Priority Heap

A priority heap is a heap-based structure used to implement priority queues. Each element is associated with a priority, and elements with higher priority are served before others. In a max priority heap, the highest priority element is always at the root. In a min priority heap, the lowest priority element is at the root. Priority heaps ensure fast insertion and removal of priority elements.

5. Applications of Priority Queue

Priority queues are widely used in computer science. They are used in job scheduling systems to execute high-priority tasks first. They are used in operating systems for process management. Priority queues are also used in graph algorithms, data compression techniques, and network routing. They help in managing tasks efficiently based on importance.

6. Max Heap and Min Heap Representation

For the data elements 3, 8, 20, 28, 42, and 54, the max heap and min heap representations are shown below.

```
Max Heap:                Min Heap:

        54                     3
       /  \                   /  \
     42    28                8    20
    /  \                    /  \
  20    8                  28  42
 /
3                         54
```

7. Difference Between Max Heap and Min Heap

A max heap is a heap in which the parent node always contains a value greater than or equal to its children, so the largest element is stored at the root. A min heap is a heap in which the parent node always contains a value less than or equal to its children, so the smallest element is stored at the root. Max heaps are used when maximum priority is required, while min heaps are used when minimum priority is required.