

# Future ready talent

## Project

NAME: Talari Uday kiran

Project name: Java heap sort

Mail id: ctalari45@gmail.com

Course: B-TECH

## Java Program for Heap Sort

Developed and created by John Gosling in 1995 in Sun Microsystems, Java is a general-purpose, object-oriented programming language. It was developed and intended to follow the WORA concept which means Write Once Run Anywhere i.e. compiled Java code can run on all platforms that support Java without the need for recompilation. Java offers various applications in the area of mobile development, web application development being the major areas. Java is one of the most popular and in-demand programming languages to learn

1. Heap sort is a comparison based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the maximum element and place the maximum element at the end. We repeat the same process for remaining element.

2. Heapsort is a popular and efficient sorting algorithm. The concept of heap sort is to eliminate the elements one by one from the heap

part of the list, and then insert them into the sorted part of the list.

3.Heap sort is a comparison based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the maximum element and place the maximum element at the end. Then repeat the same process for the remaining element. Let's write the program and understand its working.

Write a Java program to implement HeapSort Algorithm –

## Progame

```
package Edureka1;

public class HeapSort
{
    public void sort(int arr[])
    {
        int n = arr.length;

        // Build heap (rearrange array)
        for (int i = n / 2 - 1; i >= 0; i--)
            heapify(arr, n, i);

        // One by one extract an element from heap
        for (int i=n-1; i>=0; i--)
        {
            // Move current root to end
            int temp = arr[0];
            arr[0] = arr[i];
            arr[i] = temp;

            // call max heapify on the reduced heap
            heapify(arr, i, 0);
        }
    }
}
```

```
}  
  
}  
  
void heapify(int arr[], int n, int i)  
{  
    int largest = i; // Initialize largest as root  
    int l = 2*i + 1; // left = 2*i + 1  
    int r = 2*i + 2; // right = 2*i + 2  
    // If left child is larger than root  
    if (l < n && arr[l] > arr[largest])  
        largest = l;  
    // If right child is larger than largest so far  
    if (r < n && arr[r] > arr[largest])  
        largest = r;  
    // If largest is not root  
    if (largest != i)  
    {  
        int swap = arr[i];  
        arr[i] = arr[largest];  
        arr[largest] = swap;  
    }  
}
```

```
// Recursively heapify the affected sub-tree
heapify(arr, n, largest);
}
}

/* A utility function to print array of size n */
static void printArray(int arr[])
{
    int n = arr.length;
    for (int i=0; i<n; ++i)
        System.out.print(arr[i]+" ");
    System.out.println();
}

// Driver program
public static void main(String args[])
{
    int arr[] = {12, 11, 13, 5, 6, 7};
    int n = arr.length;
    HeapSort ob = new HeapSort();
    ob.sort(arr);
}
```

```
System.out.println("Sorted array is");  
printArray(arr);  
}  
}
```

Output

5,6,7,11,12,13