

Objective: Implementation of Heap Sort

Heap Sort

Heap Sort is a popular and efficient sorting algorithm in computer programming. Learning how to write the heap sort algorithm requires knowledge of two types of data structures - arrays and trees.

Heap sort works by visualizing the elements of the array as a special kind of complete binary tree called a heap.

Heap Sort Algorithm:

```
heapify(array)
    Root = array[0]
    Largest = largest( array[0] , array [2*0 + 1]. array[2*0+2])
    if(Root != Largest)
        Swap(Root, Largest)
```

How Heap Sort Works?

1. Since the tree satisfies Max-Heap property, then the largest item is stored at the root node.
2. **Swap:** Remove the root element and put at the end of the array (nth position) Put the last item of the tree (heap) at the vacant place.
3. **Remove:** Reduce the size of the heap by 1.
4. **Heapify:** Heapify the root element again so that we have the highest element at root.
5. The process is repeated until all the items of the list are sorted.

Code:

```
def heapify(arr, n, i):  
    lrgst = i  
    l = 2 * i + 1  
    r = 2 * i + 2  
    if l < n and arr[i] < arr[l]:  
        lrgst = l  
    if r < n and arr[lrgst] < arr[r]:  
        lrgst = r  
    if lrgst != i:  
        arr[i], arr[lrgst] = arr[lrgst], arr[i]  
        heapify(arr, n, lrgst)  
  
def heapSort(arr):  
    n = len(arr)  
    for i in range(n//2, -1, -1):  
        heapify(arr, n, i)  
    for i in range(n-1, 0, -1):  
        arr[i], arr[0] = arr[0], arr[i]  
        heapify(arr, i, 0)  
    return arr  
  
arr = [1, 12, 9, 5, 6, 10]  
print("Sorted array is", heapSort(arr))
```

```
6_Heap_Sort.py - C:\Users\Triya Gaur\Desktop\New folder\Iird year\Vth Sem\DAA Lab\6_Heap_Sort.py (3.7.3)
File Edit Format Run Options Window Help

def heapify(arr, n, i):
    lrgst = i
    l = 2 * i + 1
    r = 2 * i + 2
    if l < n and arr[l] < arr[i]:
        lrgst = l
    if r < n and arr[r] < arr[lrgst]:
        lrgst = r
    if lrgst != i:
        arr[i], arr[lrgst] = arr[lrgst], arr[i]
        heapify(arr, n, lrgst)

def heapSort(arr):
    n = len(arr)
    for i in range(n//2, -1, -1):
        heapify(arr, n, i)

    for i in range(n-1, 0, -1):
        arr[i], arr[0] = arr[0], arr[i]
        heapify(arr, i, 0)
    return arr

arr = [1, 12, 9, 5, 6, 10]
heapSort(arr)
n = len(arr)
print("Sorted array is", heapSort(arr))

|
```

Output:

Sorted array is [1, 5, 6, 9, 10, 12]

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Triya Gaur\Desktop\New folder\Iird year\Vth Sem\DAA Lab\6_Heap_Sort.py
Sorted array is [1, 5, 6, 9, 10, 12]
>>> |
```

Time Complexities:

- **Worst Case Complexity:** $O(n \log n)$
- **Best Case Complexity:** $O(n \log n)$
- **Average Case Complexity:** $O(n \log n)$

Heap Sort Applications:

- Heap sort is not a Stable sort, and requires a constant space for sorting a list.
- Heap Sort is very fast and is widely used for sorting.