Algorithm Analysis and Data Structures: Assignment 3 – Heap Sort

Implement the heap sort.

Given a random list of numbers in an array (index zero stores the number of nodes).

Sort them in ascending order.

1. You must first make a heap and then sort it.

Initialize an array of random numbers (at least 20)

Print the list.

Make a heap

Print the list

Sort it

print the list again.

Code:

import java.util.\*;

public class HeapSort {

public static void main(String[] args)

{

HeapSort hs = new HeapSort();

Random r = new Random();

int[] arr = new int[21];

arr[0] = arr.length - 1;

for(int i=1; i< arr.length; i++)

{

arr[i] = r.nextInt(50);

}

System.out.println("\nInitial list of elements in the array, where the first element in the array is equal to the number of nodes");

hs.PrintArray(arr);

hs.Heapify(arr);

System.out.println("Array after converting into a heap, where the first element in the array is equal to the number of nodes");

hs.PrintArray(arr);

hs.HeapSort(arr);

System.out.println("Heap after sorting, where the first element in the array decreases by 1 in every pass of sorting and becomes 1 by the time the heap is sorted");

hs.PrintArray(arr);

}

/\*Heapify:

this method is used to heapify from n/2th node as the nodes after that are leaf nodes\*/

public void Heapify(int [] arr)

{

int n = arr[0];

for(int i = n/2; i>0;i--)

{

MaxHeapify(arr,n,i);

}

}

/\*SwapMaximun

this method is used to remove the maximum element from the first position and place it at the appropriate position based on the value of a[0] \*/

public void SwapMaximun(int[] arr)

{

int counter = arr[0];

int temp = arr[counter];

arr[counter] = arr[1];

arr[1] = temp;

arr[0] = arr[0] - 1;

}

/\* HeapSort:

this method is used to heapify and then swap the maximum and by the end the heap would be sorted\*/

public void HeapSort(int [] arr)

{

while (arr[0] >1)

{

Heapify(arr);

SwapMaximun(arr);

}

}

/\*MaxHeapify:

this method is used to set the MaxHeap property\*/

public void MaxHeapify(int[] arr, int n, int i)

{

int large= i;

int left = 2\*i;

int right = (2\*i)+1;

if(left <= n && arr[left] > arr[large])

large = left;

if(right <= n && arr[right] > arr[large])

large = right;

if(large != i)

{

int temp = arr[i];

arr[i] = arr[large];

arr[large]= temp;

MaxHeapify(arr,n, large);

}

}

/\*PrintArray:

this method is used to print the elements in the array\*/

public void PrintArray(int[] arr)

{

for(int i=0; i< arr.length; i++)

System.out.print(" "+arr[i]);

System.out.println("\n");

}

}

Instructions to compile:

1. Create a .java file in Sublime text or any text editor by the name HeapSort.java
2. Place the code in HeapSort.java and save
3. Open command prompt and redirect to the path where HeapSort.java is saved
4. Use javac HeapSort.java command, to compile the code
5. After successful compilation use java HeapSort command, to run the code and see the output

Screen Shot of the output

