

**LEBANESE AMERICAN UNIVERSITY**  
**School of Arts and Science**  
**Department of Computer Science and Mathematics**

**CSC 310: Algorithms and Data Structures**

---

**Lab V**

---

**Important Note:** Input should be read from a file named after the problem name. For example, problem 1 has the input file “problem1.in”. You are also required to test your programs on an array of 100,000 elements randomly generated so you can output the running time.

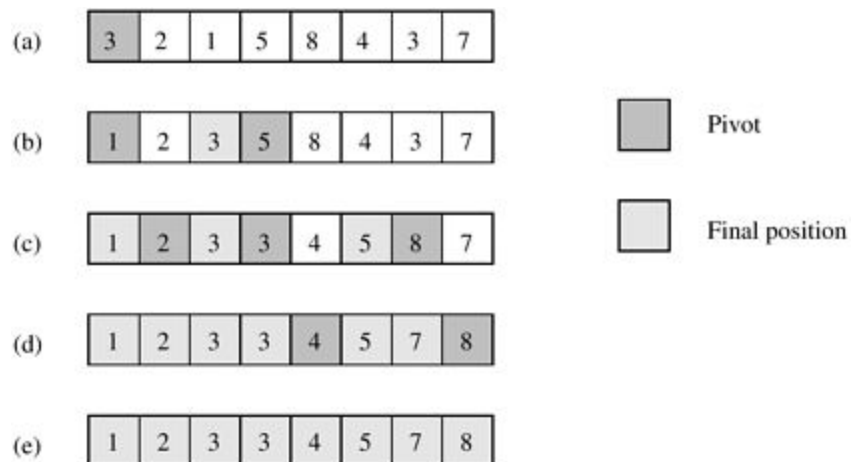
**Output Running Time:**

```
long t1 = System.currentTimeMillis();  
// Your Sorting Algorithm  
long running_time = System.currentTimeMillis() - t1;
```

**Problem 1**

Given an array of integers, write a program that sorts the array using Quicksort.

Recursively keep on dividing the array into subarrays by selecting a pivot and placing it in its right position (elements on the left of the pivot are smaller and elements on the right of the pivot are larger) until no more subdivision can be made. For this problem the pivot will be the leftmost element. Given a sequence:



### **Input**

Each test case is made up of two lines. The first line has an integer N representing the number of elements in the array. The second line contains N integers representing the values in the array.

### **Output**

For each test case, output the sorted array.

#### **Sample Input**

```
3
7
25 13 10 30 15 27 37
4
6 7 8 9
6
10 7 15 13 4 6
```

#### **Sample Output**

```
10 13 15 25 27 30 37
6 7 8 9
4 6 7 10 13 15
```

### **Problem 2**

Given an array of integers, write a program that sorts the array using Quicksort.

Recursively keep on dividing the array into subarrays by selecting a pivot and placing it in its right position(elements on the left of the pivot are smaller and elements on the right of the pivot are larger) until no more subdivision can be made. For this problem the pivot will be the median(middle value) of three values which are the first, last, and middle elements.

### **Input**

Each test case is made up of two lines. The first line has an integer N representing the number of elements in the array. The second line contains N integers representing the values in the array.

### **Output**

For each test case, output the sorted array.

#### **Sample Input**

```
3
7
25 13 10 30 15 27 37
4
6 7 8 9
6
```

#### **Sample Output**

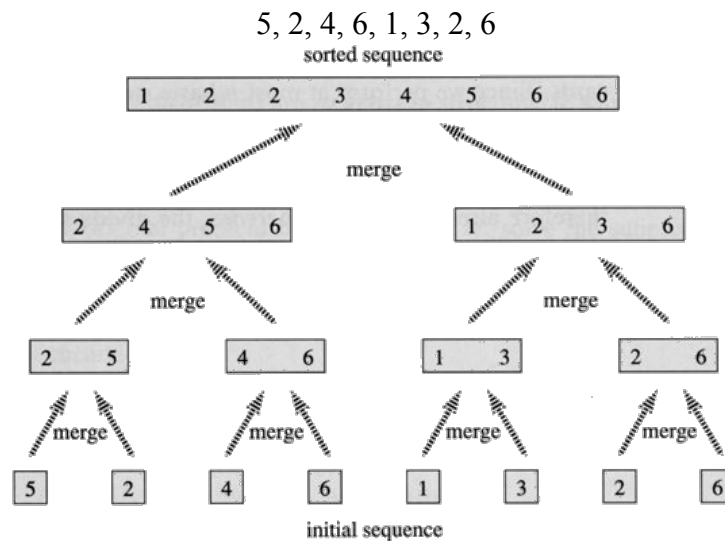
```
10 13 15 25 27 30 37
6 7 8 9
4 6 7 10 13 15
```

10 7 15 13 4 6

### **Problem 3**

Given an array of integers, write a program that sorts the array using Merge Sort.

Recursively keep on dividing the array into subarrays until you reach a single element (a single element is considered sorted). After that keep on merging subarrays to produce a new subarray until 1 subarray remains which will be the sorted array. Given a sequence:



### **Input**

Each test case is made up of two lines. The first line has an integer N representing the number of elements in the array. The second line contains N integers representing the values in the array.

### **Output**

For each test case, output the sorted array.

#### **Sample Input**

3  
7  
25 13 10 30 15 27 37  
4  
6 7 8 9  
6  
10 7 15 13 4 6

#### **Sample Output**

10 13 15 25 27 30 37  
6 7 8 9  
4 6 7 10 13 15

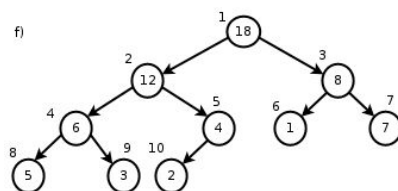
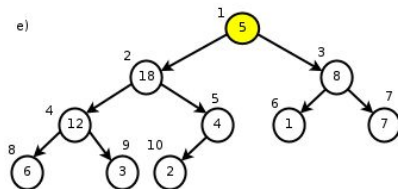
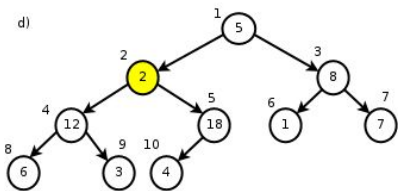
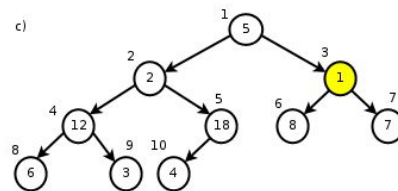
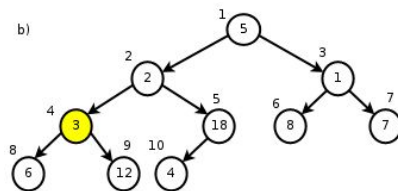
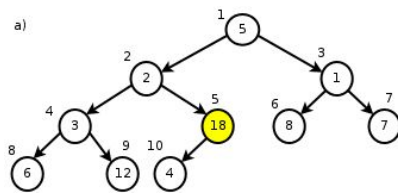
### **Problem 4**

Implement a Max-heap data structure (the parent is always larger than its children) using an array. More specifically, you are asked to implement the insert and heapify method.

**Hint:** In the array representation, the root is located at index 0. Moreover, for any node at index  $i$ , the left child is always at index  $2 * \text{parentIndex} + 1$  and the right at  $2 * \text{parentIndex} + 2$ .

For example given an array of length 10:

5, 2, 1, 3, 18, 8, 7, 6, 12, 4



### **Input**

Each test case is made up of two lines. The first line has an integer  $N$  representing the number of elements in the array. The second line contains  $N$  integers representing the values in the array.

### **Output**

For each test case, the heap array.

#### **Sample Input**

```
2
5
3 2 1 5 4
4
6 7 8 9
6
10 7 15 13 4 6
```

#### **Sample Output**

```
5 4 3 2 1
9 8 6 7
15 13 7 10 4 6
```

### **Problem 5**

Given an array of integers, write a program that sorts the array using Heap Sort.

Using the above Max-heap structure you implemented, you can sort the array of length  $n$  as follows:

1. Swap the first element(Largest element) in the array with the  $n^{\text{th}}$  position .
2. Heapify the array on the  $n - 1$  elements.
3. Decrement  $n$  until  $n$  becomes 0.

### **Input**

Each test case is made up of two lines. The first line has an integer  $N$  representing the number of elements in the array. The second line contains  $N$  integers representing the values in the array.

### **Output**

For each test case, output the sorted array.

#### **Sample Input**

```
2
7
25 13 10 30 15 27 37
4
```

#### **Sample Output**

```
10 13 15 25 27 30 37
6 7 8 9
```

