Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Institut für Theoretische Informatik
Peter Widmayer
Sandro Montanari
Tobias Pröger

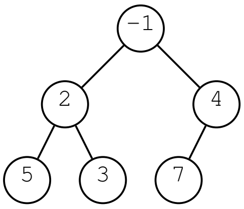
6th March 2013

Datenstrukturen & Algorithmen Programming Exercise 3 FS 13

In this exercise we will implement a binary *min-heap* using an array. The following operations should be supported:

- `Insert`(v) inserts the element v into the heap and restores the heap property.
- `Extract-Min` extracts the minimum from the heap and returns it.
- `Query-Last` returns the element that is stored in the last position of the array representing the heap.

If, e.g., the numbers $4, 5, 2, 3, -1, 7$ are inserted (in this order) into an initially empty heap, then we obtain the following heap:



This heap is represented by the array $-1, 2, 4, 5, 3, 7$. Therefore, the `Query-Last` operation returns `7` for the above heap.

Input The first line of the input contains only the number t of test instaces. After that, we have exactly one line for each test instance containing the numbers $n, v_1, v_2, ..., v_n$. While $n \in \mathbb{N}$, $1 \leq n \leq 1000$ describes the number of following integers, $v_i \in \mathbb{Z}$, $-1000 \leq v_i \leq 1000$ is the next element to be inserted. Every test instance starts with an empty heap, and the operations `Insert`(v_1), `Query-Last`, `Insert`(v_2), `Query-Last`, ..., `Insert`(v_n), `Query-Last` are executed in exactly that order.

Output For every test instace, we want to output two lines. The first one contains the output of all n `Query-Last` operations. The second one contains the output of n succeeding `Extract-Min` operations. This especially means that the first `Extract-Min` operation is performed *after* all `Query-Last` operations were executed.

Example

Input:

```
2
3 1 2 3
5 5 7 3 4 2
```

Output:

```
1 2 3
1 2 3
5 7 5 7 4
2 3 4 5 7
```

Hand-in: until Wednesday, 13th March 2013.