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

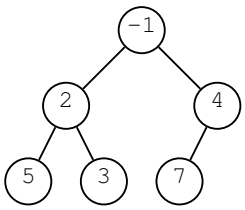
6. März 2013

Datenstrukturen & Algorithmen Programmieraufgabe 3 FS 13

In dieser Aufgabe wollen wir einen binären *Min-Heap* mithilfe eines Arrays implementieren. Konkret sollen dabei die folgenden Operationen unterstützt werden:

- `Insert`(v) fügt das Element v in den Heap ein und stellt die Heap-Bedingung wieder her.
- `Extract-Min` extrahiert das Minimum aus dem Heap und liefert es zurück.
- `Query-Last` liefert das Element in der letzten Position des Arrays zurück, das den Heap repräsentiert.

Werden beispielsweise die Zahlen $4, 5, 2, 3, -1, 7$ (in dieser Reihenfolge) in einen initial leeren Heap eingefügt, dann ergibt sich der folgende Heap:



Dieser wird durch das Array $-1, 2, 4, 5, 3, 7$ repräsentiert. Die Operation `Query-Last` liefert für den obigen Heap also `7` zurück.

Eingabe Die erste Zeile der Eingabe enthält lediglich die Anzahl t der Testinstanzen. Es folgt eine Zeile für jede Testinstanz mit den Zahlen $n, v_1, v_2, ..., v_n$. Dabei ist $n \in \mathbb{N}$, $1 \leq n \leq 1000$, die Anzahl der folgenden Elemente und $v_i \in \mathbb{Z}$, $-1000 \leq d_i \leq 1000$, das nächste einzufügende Element. Jede Testinstanz startet mit einem leeren Heap, und die Operationen `Insert`(v_1), `Query-Last`, `Insert`(v_2), `Query-Last`, ..., `Insert`(v_n), `Query-Last` werden in genau dieser Reihenfolge ausgeführt.

Ausgabe Für jede Testinstanz sollen zwei Zeilen ausgegeben werden. Die erste enthält die Ausgaben der n `Query-Last`-Operationen. Die zweite enthält die Ausgabe, wenn *danach* (also erst nachdem alle `Query-Last`-Operationen ausgeführt wurden) die Operation `Extract-Min` n Mal ausgeführt wird.

Beispiel

Eingabe:

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

Ausgabe:

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

Abgabe: Bis Mittwoch, den 13. März 2013.