

## Eigenschaften

- benutzt [[Heap]] als Datenstruktur
- nutzt, dass Maximum immer an erster Stelle steht

```

HEAPSORT(A)
1:  BAUE_HALDE(A)
2:  FOR i ← n-1 DOWNTO 1
3:    VERTAUSCHE(A[0], A[i])
4:    N ← N-1
5:    VERHALDE(A,0)
    
```

N ≤ n aktuelle Haldengröße:

$$\begin{array}{c}
 \text{BAUE\_HALDE} \quad \text{VERHALDE} \\
 \swarrow \quad \searrow \\
 T(n) \in O(n) + (n-1) O(\log n) \\
 = O(n \log n)
 \end{array}$$

•

## Warteschlangen-Problem

- Warteschlange mit Priorisierung
  - anders als [[Queue]]

Eine Warteschlange ist eine dynamische Datenstruktur die auf einer Menge A folgende Operationen erlaubt:

- EINFÜGE (A, x)  $A = A \cup \{x\}$
- MAXIMUM(A) return max A
- ENTFERNE-MAX(A)  $A = A \setminus \{ \max A \}$

```

MAXMUM(A)
1:  return A[0]
    
```

O(1) Zeit

```

ENTFERNE-MAX(A)
1:  A[0] ← A[N-1]
2:  N ← N-1
3:  VERHALDE(A, 0)
    
```

O(log n) Zeit

```

EINFÜGE (A, x)
1:  n ← n+1, A[ n-1 ] ← x, i ← n
2:  WHILE i>0 AND A[ i ] > A[ Parent(i) ] DO
3:    Vertausche ( A[i], A[ Parent(i) ] )
4:    i ← Parent(i)
    
```

O(log n) Zeit

•