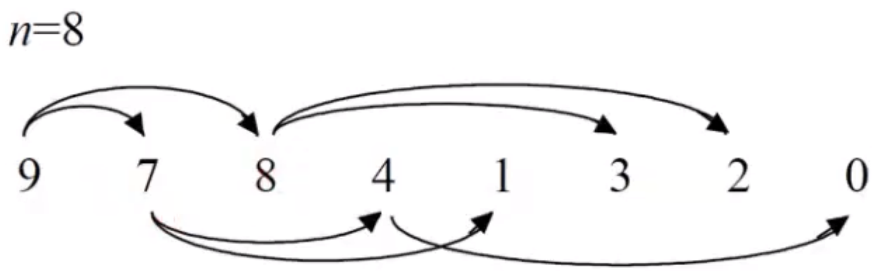
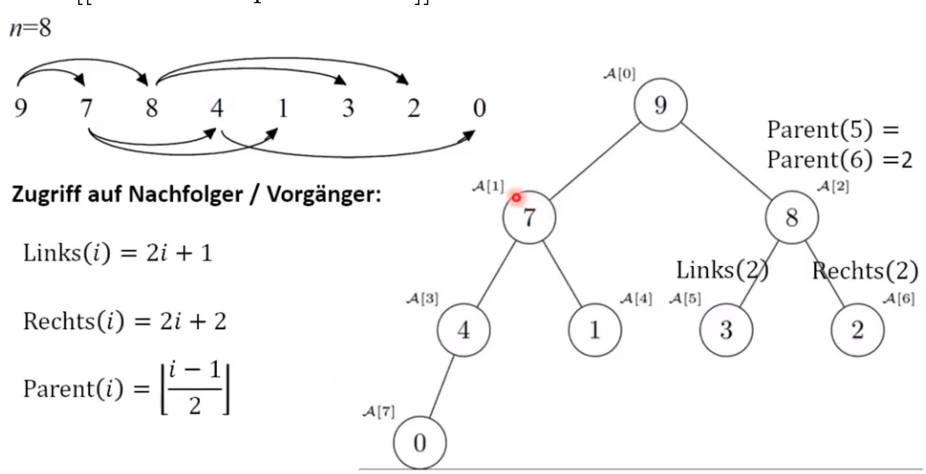
Definition

- Heap wird auch Halde genannt
  Definition (max-heap):
- Eine Halde (Heap) ist ein **lineares Feld** A[0..n-1], wobei gilt:
  A[i] ≥ max {A[2i+1], A[2i+2]}, für i=0,1,...,⌊n/2⌋-1 (Haldenbedingung)
- 

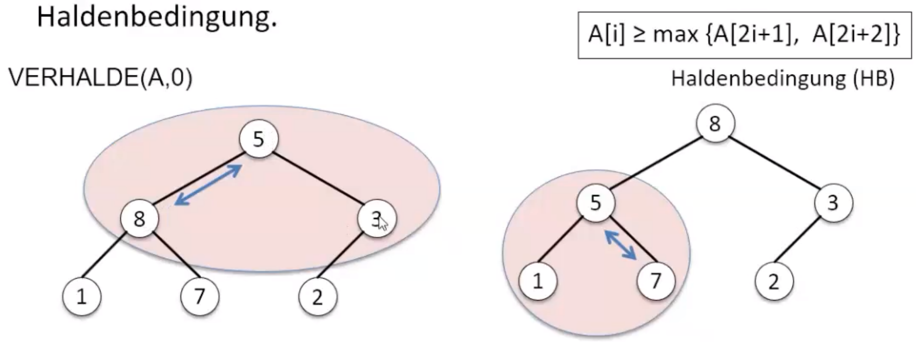
- nicht automatisch absteigend sortiert
- Darstellung als Graph/Binärbaum
  - siehe [[Bäume & Spannbäume]]
  - 

- Eigenschaften
  - $A[0]$ ist Maximum (Wurzel)
  - vollständiger Baum
    - letzte Ebene evtl. nicht komplett
  - jeder Teilbaum wieder Halde
  - $h = \lfloor log_2(n) \rfloor$

Heapify

- Verhalde-Prozedur

Verhalden von Element A[i]
Voraussetzung: Die Teilbäume mit Wurzel LINKS(i) und RECHTS(i) sind Halden, aber Element i verletzt möglicherweise die Haldenbedingung.



–

```
VERHALDE(A, i)
//N…aktuelle Haldengröße
l ← LINKS(i), r ← RECHTS(i)
index ← i
IF l<N and A[l]>A[i] THEN index←l
IF r<N and A[r]>A[index] THEN index←r
IF i≠index THEN
    vertausche A[i], A[index]
    VERHALDE (A, index)
```

index: Index von max{A[i],A[l],A[r]}

– **Laufzeit: T(n) = O(log n)**

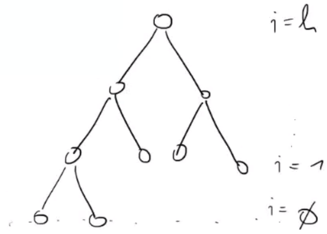
- Aufbau einer Halde mittel Heapify
  - gegeben lineares Feld in beliebiger Reihenfolge
  - Blätter (einzelnes Element) sind triviale Halden
  - Verhalde auf Eltern der Blätter (vorletzte Schicht) anwenden
  - Wiederholen für alle Knoten bis zur Wurzel
  - ```
    BAUE_HALDE(A)
    FOR i ← ⌊n/2⌋-1 DOWNTO 0
        VERHALDE (A, i)
    ```
  - Laufzeit
    ```

```
BAUE_HALDE(A)
FOR i ← ⌊n/2⌋-1 DOWNTO 0
    VERHALDE (A, i)
```

- **Naive Analyse:** n/2 * VERHALDE
  Laufzeit: $T(n) \in \frac{n}{2} O(\log n) \in O(n \log n)$

- **Aber:** Element der Höhe *h* kann in O(*h*) Zeit verhaldet werden ⇒ Laufzeit $\boldsymbol{T(n) \in O(n)}$

*



#Elemente 2^{h-i}

$h = \lfloor \log n \rfloor$
$n \geq 2^h$

$$T(n) \leq \sum_{i=0}^{h} i\, 2^{h-i} = \sum_{i=0}^{h} i \cdot \frac{\overset{\leq n}{2^h}}{2^i} \leq n \cdot \overbrace{\sum_{i=0}^{h} i \cdot \left(\frac{1}{2}\right)^i}^{\leq 2}$$

*