

1. Du kannst jede der beiden Darstellungen eines Heaps (als Liste ohne „nulltes“ Element oder als vollständigen Binärbäum) in die jeweils andere Form transformieren.
2. Du kannst in der Listendarstellung aus dem Index eines Kindknotens den Index des zugehörigen Elternknotens berechnen und umgekehrt.
3. Du kannst erkennen, ob ein Heap eine der beiden „schwachen“ Ordnungsstrukturen (Min-Heap oder Max-Heap) aufweist.
4. Du kannst die folgenden Operationen auf einem Min-Heap oder Max-Heap aufzählen, ihre Laufzeitkomplexität angeben und in der Baumdarstellung schrittweise visualisieren (dafür wird ein Raster zur Verfügung gestellt):
 - `heap.insert(key)`
 - `heap.delMin()` bzw. `heap.delMax()`

Beachte, dass neben der eigentlichen Operation auch noch die Min-Heap- bzw. Max-Heap-Eigenschaft mit den Hilfsoperationen `heap.swim(i)` oder `heap.sink(i)` wieder hergestellt werden muss.