

Intro: Datastruktur, komplet/næsten komplet binært træ. Array repræsentere træet med 2 attributer A.length og A.heap-size som viser hvor mange elementer fra heap der er stored i A.length.

Kan finde parent og childs ved indexes.  
heap-property- 2 kinds: max heap og min heap.

$h = \theta(\log(n))$  longest simple path.  
Basic operations in  $O(\lg n)$  tid.  
max-heapify(A,i) -  $O(\lg n)$  Kører kun med udvalgt element  
build-max-heap(A) - linear time - Vis Køretid.  
Heapsort -  $O(n \lg n)$  - In plac, fylder ikke meget

Prioritetskøer - datastruktur til at holde et sæt S elementer med en key.  
max-priotets kø understøtter:  
Extract-Max(S) -  $O(\lg n)$   
Increase-Key(S,x,k) -  $O(\lg n)$   
Insert(S,x) -  $O(\lg n)$   
Maximum(S)

Altså kan heaps lave operationer på priotetskøer i  $O(\lg n)$  tid, hvor  $n$ =noder.

Find ud af loop invariant.  
A[9,7,5,4,2,1]