Computer Science 331

Heap Sort

Mike Jacobson

Department of Computer Science
University of Calgary

Lecture #26

Outline

HeapSort

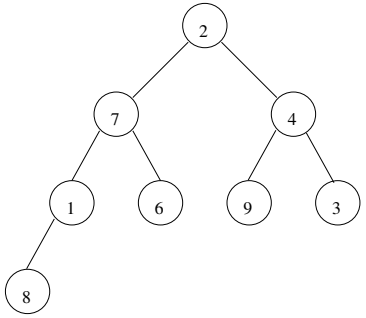
Idea:

1. Use Build-Max-Heap to convert the input array into a representation of a heap
2. Repeatedly use Delete-Max to extract the largest element in the unsorted part of the array and move it into position

Pseudocode

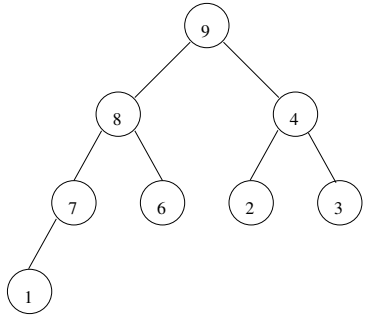
Heap Sort(A)
  $n =$ heap-size(A)
  if $n > 1$ **then**
    Build-Max-Heap(A)
    $i = n - 1$
    while $i > 0$ **do**
      $largest =$ **Delete-Max**(A)
      $A[i] = largest$
      $i = i - 1$
    end while
  end if

Example



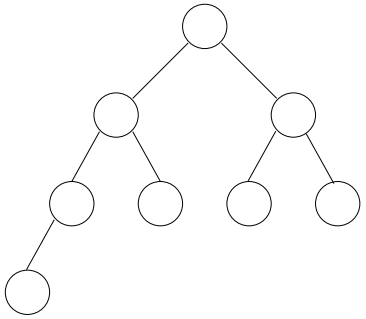
0	1	2	3	4	5	6	7
2	7	4	1	6	9	3	8

Example: After Build-Max-Heap



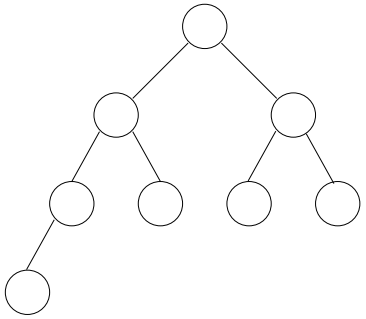
0	1	2	3	4	5	6	7
9	8	4	7	6	2	3	1

$A[7] =$ **Delete-Max**(A)



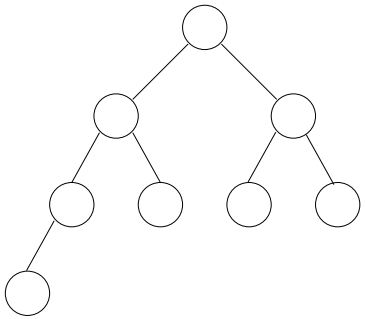
0	1	2	3	4	5	6	7

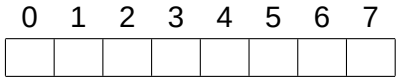
$A[6] =$ **Delete-Max**(A)



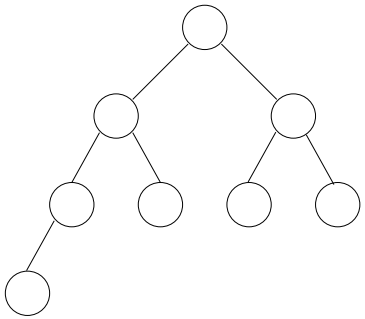
0	1	2	3	4	5	6	7

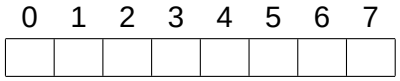
$A[5] =$ **Delete-Max**(A)



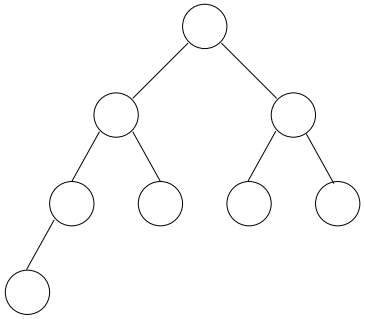


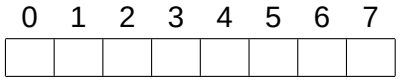
$A[4] =$ **Delete-Max**(A)



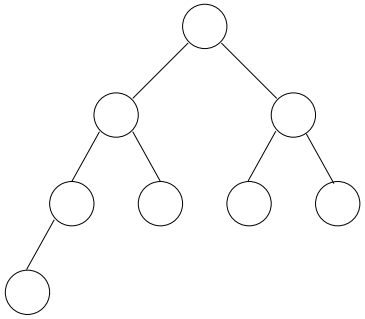


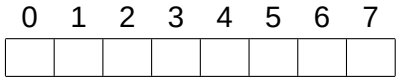
$A[3] =$ **Delete-Max**(A)



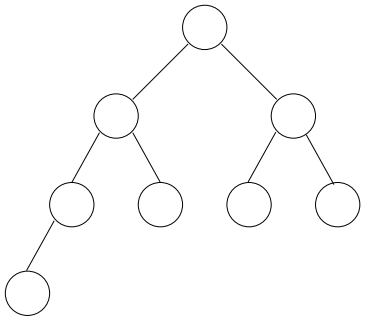


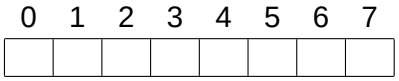
$A[2] =$ **Delete-Max**(A)





$A[1] =$ **Delete-Max**(A)





Partial Correctness

Loop invariant: if the loop is executed at least k times then, after the kth iteration:

- $0 \leq i \leq n-1$, and $i = n-1-k$
- heap-size$(A) = i+1 = n-k$
- A represets a max-heap with size $i+1$
- the last $n-i-1$ entries are all greater than or equal to each of the first $i+1$ entries in A
- $A[j] \leq A[j+1]$ for $i \leq j \leq n-2$
- the entries of A are reordered but otherwise unchanged

If true and loop terminates, then $i = 0$ and

-
-

Termination and Efficiency

Loop variant: $f(n,i) = i$

-
-
-

Worst-case cost of **heapSort**:

-
-

Advantage over **mergeSort**:

References

Textbook, Section 10.8

- A simplified presentation: “Max-Heapify” is not separately analyzed
- Build-Max-Heap slightly different (starts at index 0)

Cormen, Leiserson, Rivest and Stein, *Introduction to Algorithms*, Second Edition, Chapter 6: A discussion of Heap Sort that is closer to the presentation in these notes.