

CSI 503 – Data Structures and Algorithms

Pseudocode Descriptions of Heap Operations

Handout 7.1

Note: This handout is referred to in the slides for Lecture 7.

(a) Pseudocode for Heapify:

Heapify (A, i)

```
1. l = Left(i);  r = Right(i)

// Find the index (m) of the largest value among
// A[i], A[l] and A[r].

2. if ( (l <= heap_size(A) and (A[l] > A[i])) )
    then m = l
    else m = i
3. if ((r <= heap_size(A) and (A[r] > A[m]))
    then m = r

// Swap A[i] with A[m] and use recursion.

4. if (m != i)
    then
        4.1 Exchange A[i] with A[m].
        4.2 Heapify(A,m)
```

(b) Pseudocode for Build-Heap:

Build-Heap (A)

```
1. heap_size[A] = length[A]; n = length[A]

2. for i = floor(n/2) downto 1 do
    Heapify(A, i)
```

(over)

(c) Pseudocode for Heapsort:

Heapsort(A)

1. $n = \text{length}[A]$
 2. Build-Heap(A)
 3. for $i = n$ downto 2 do
 - 3.1 Exchange $A[1]$ with $A[i]$
 - 3.2 $\text{heap_size}[A] = \text{heap_size}[A] - 1$
 - 3.3 Heapify(A,1)
-

(d) Pseudocode for Heap-Extract-Max:

Heap-Extract-Max(A)

1. if ($\text{heap_size}[A] = 0$)
 then print "Error: Heap underflow" and stop.
 2. $\text{max} = A[1]$
 3. $A[1] = A[\text{heap_size}[A]]$
 4. $\text{heap_size}[A] = \text{heap_size}[A] - 1$
 5. Heapify(A,1)
-

(e) Pseudocode for Heap-Increase-Key:

Heap-Increase-Key(A, x, k) // The value of $A[x]$ must be changed to k.

1. if ($k < A[x]$)
 then print "Error: New key value smaller than current" and stop.
 2. $A[x] = k$
 3. while (($i > 1$) and ($A[\text{Parent}(i)] < A[i]$)) do
 - 3.1 Exchange $A[i]$ and $A[\text{Parent}(i)]$
 - 3.2 $i = \text{Parent}(i)$
-

(f) Pseudocode for Heap-Insert:

Heap-Insert(A, k) // k : Key to be inserted.

1. $\text{heap_size}[A] = \text{heap_size}[A] + 1$ // Must do a heap overflow check here.
 2. $A[\text{heap_size}[A]] = -\text{infinity}$
 3. Heap-Increase-Key(A, $\text{heap_size}[A]$, k)
-