CSI 503 – Data Structures and Algorithms
Pseudocode Descriptions of Heap Operations

(a) Pseudocode for Heapify:

Heapify \( A, i \)

1. \( l = \text{Left}(i); \ r = \text{Right}(i) \)

// Find the index (m) of the largest value among

2. if \((l \leq \text{heap\_size}(A) \text{ and } (A[l] > A[i]))\)
then \( m = l \)
else \( m = i \)

3. if \((r \leq \text{heap\_size}(A) \text{ and } (A[r] > A[m]))\)
then \( m = r \)

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

4. if \((m \neq i)\)
then
   4.1 Exchange \( A[i] \) with \( A[m] \).
   4.2 Heapify\((A, m)\)

(b) Pseudocode for Build-Heap:

Build-Heap \((A)\)

1. \( \text{heap\_size}[A] = \text{length}[A]; \ n = \text{length}[A] \)

2. for \( i = \text{floor}(n/2) \) downto 1 do
   Heapify\((A, i)\)
(c) Pseudocode for Heapsort:

Heapsort(A)

1. n = length[A]
2. Build-Heap(A)
3. for i = n downto 2 do
   3.2 heap_size[A] = heap_size[A]-1
   3.3 Heapify(A,1)

(d) Pseudocode for Heap-Extract-Max:

Heap-Extract-Max(A)

1. if (heap_size[A] = 0)
   then print "Error: Heap underflow" and stop.
2. max = A[1]
4. heap_size[A] = 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[Parent(i)] < A[i]) ) do
   3.1 Exchange A[i] and A[Parent(i)]
   3.2 i = Parent(i)

(f) Pseudocode for Heap-Insert:

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

2. A[heap_size[A]] = -infinity
3. Heap-Increase-Key(A, heap-size[A], k)