

**WIA2005 Algorithm Design & Analysis**  
**Semester 2, 2016/17**  
**Lab 6**

---

1. Implement the MAX-HEAPIFY algorithm.

```
MAX-HEAPIFY(A, i)
1  l = LEFT(i)
2  r = RIGHT(i)
3  if l ≤ A.heap-size and A[l] > A[i]
4      largest = l
5  else largest = i
6  if r ≤ A.heap-size and A[r] > A[largest]
7      largest = r
8  if largest ≠ i
9      exchange A[i] with A[largest]
10     MAX-HEAPIFY(A, largest)
```

Given an array  $A = (11, 4, 74, 55, 3, 17, 8, 46, 43, 33)$ , implement BUILD-MAX-HEAP using MAX-HEAPIFY to generate the max heap.

```
BUILD-MAX-HEAP(A)
1  A.heap-size = A.length
2  for i =  $\lfloor A.length/2 \rfloor$  downto 1
3      MAX-HEAPIFY(A, i)
```

2. Implement Heapsort and sort array  $A = (11, 4, 74, 55, 3, 17, 8, 46, 43, 33)$ .

```
HEAPSORT(A)
1  BUILD-MAX-HEAP(A)
2  for i = A.length downto 2
3      exchange A[1] with A[i]
4      A.heap-size = A.heap-size - 1
5      MAX-HEAPIFY(A, 1)
```

3. Implement the Priority Queue containing the following operations.  
Demonstrate the operation using the the input array in the question (1).

```
HEAP-MAXIMUM(A)
1  return A[1]
```

```
HEAP-EXTRACT-MAX(A)
1  if A.heap-size < 1
2      error "heap underflow"
3  max = A[1]
4  A[1] = A[A.heap-size]
5  A.heap-size = A.heap-size - 1
6  MAX-HEAPIFY(A, 1)
7  return max
```

HEAP-INCREASE-KEY( $A, i, key$ )

```
1  if  $key < A[i]$ 
2      error "new key is smaller than current key"
3   $A[i] = key$ 
4  while  $i > 1$  and  $A[PARENT(i)] < A[i]$ 
5      exchange  $A[i]$  with  $A[PARENT(i)]$ 
6       $i = PARENT(i)$ 
```

MAX-HEAP-INSERT( $A, key$ )

```
1   $A.heap-size = A.heap-size + 1$ 
2   $A[A.heap-size] = -\infty$ 
3  HEAP-INCREASE-KEY( $A, A.heap-size, key$ )
```