Workshop09

Name:____________________________________

Note: This assignment will be collected at the due date and time PRIOR to the midterm and may NOT be completed late. The `max_heap.cpp` OR `max_heap.py` file collected at that time will be the one graded. There will be no late submissions for this assignment.

1. **Using Figure 6.3 as a model**, illustrate the operation of BUILD-MAX-HEAP on the array `A = {5,3,17,10,84,19,6,22,9}`. Write your complete solution on paper or in lucidchart.

2. In CoCalc Workshop `6.3-Max_Heap`, write a program `max_heap.cpp` OR `max_heap.py` that ultimately implements and demonstrates the BUILD-MAX-HEAP(A,n) pseudocode on page 157.

Specifications: Code and utilize the following functions:

Listing 1: print vector

```
/* print_vector(v) for max_heap.cpp ONLY
*  takes integer vector v as a const reference parameter
*  Prints the contents of vector v.  v is not modified
*/
```

Listing 2: print heap

```
/* print_heap(v)
*  takes integer vector (list) v as a const reference parameter
*  Prints the contents of vector (list) v AFTER the initial unused position
*  v is not modified
*/
```

Listing 3: parent

```
/* parent(i)
*  return the index of the parent of node i
*/
```

Listing 4: left child

```
/* left(i)
*  return the index of the left child of node i
*/
```

Listing 5: right child

```
/* right(i)
*  return the index of the right child of node i
*/
```

Listing 6: max heapify

```
/*
* max_heapify(A, i, n)
* Takes a heap/vector(list) A[1..n] of size n and an index i into the array
* MAX-HEAPIFY assumes that the binary trees rooted at LEFT[i] RIGHT[i] are
* max-heaps, but that A[i] might be smaller than its children, thus violating the
* max-heap property.
*  MAX-HEAPIFY lets the value at A[i] " oat "down in the max-heap so that the
* subtree  rooted at index i obeys the max-heap property
* A is modified
*/
```

Listing 7: build max heap

```
/*
* build_max_heap(A, n)
* Takes an unordered vector A[1..n] of size n and produces a max-heap
* A is modified
*/
```

Listing 8: main

```
/*
* main()
*   Demonstrate max_heapify(A, 2, 10)on the vector/list
*   A = {-1000,16,4,10,14,7,9,3,2,8,1} and build-max-heap(A,n)
*/
```

Listing 9: build max heap example run

```
Run max-heapify on : A = {-1000,16,4,10,14,7,9,3,2,8,1}
Heap A = <16,4,10,14,7,9,3,2,8,1>

After max-heapify on : A = {-1000,16,14,10,8,7,9,3,2,4,1}
Heap A = <16,14,10,8,7,9,3,2,4,1>

Run build_max_heap on : B = {-1000,4,1,3,2,16,9,10,14,8,7}
Heap B = <4,1,3,2,16,9,10,14,8,7>

After build_max_heap on : B = {-1000,16,14,10,8,7,9,3,2,4,1}
Heap B = <16,14,10,8,7,9,3,2,4,1>
```