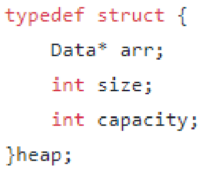
Assignment for COSE214 – 01(Algorithm class)

2019320040 Bae Hunsang(배훈상)

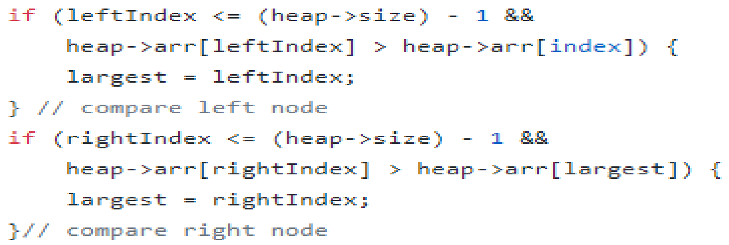
A priority queue is a data structure based on a heap. Similar to other data structures such as 'stack' or 'queue', on the priority queue, there is a particular order to manage input and output data. To make that order, the data in priority queue has some weight called **a key**. Due to the key, the user can manage the most important data as the first out data. For example, let's suppose a surgeon needs to manage the data of the patient. Then, he might want to perform surgery in urgent order. In this case, the priority queue can be a solution to manage the patients because the most ominous patient has the first order in the priority queue.



```
typedef struct {
    Data* arr;
    int size;
    int capacity;
}heap;
```

[1]

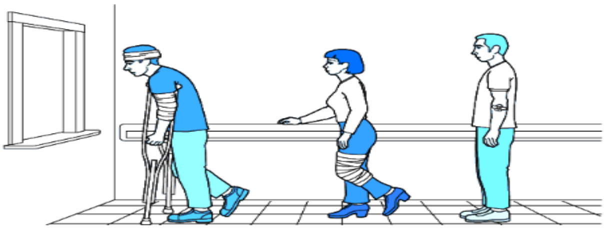
Let's suppose the heap is created based on the array. To maintain the priority queue, generally four-function is needed {Insert(), Maximum(), Extract Max(), Increase Key()}. The **Insert** function works similar to 'push' in the stack. It inserts the data into the data structure. However, unlike the stack, the Insert function sorts the array to the heap with **Heapify** function in it. Also, the user can define the compare function to manage various data types or to decide which max-heap or min-heap is created.



```
if (leftIndex <= (heap->size) - 1 &&
    heap->arr[leftIndex] > heap->arr[index]) {
    largest = leftIndex;
} // compare left node
if (rightIndex <= (heap->size) - 1 &&
    heap->arr[rightIndex] > heap->arr[largest]) {
    largest = rightIndex;
}// compare right node
```

[2]

The maximum function is quite simple. It returns the data which has the largest key value. In this case, the array's first element is returned. **Extract max** is similar to 'pop' in the stack. It extracts the most important data from the priority queue and returns that value. The same as Insert function, it also sorts the array to the heap after extraction. Finally, **Increase Key** function can change the key value of the element at the input index. Let's go to the surgeon's example. In the case that some patient's condition changes suddenly to the emergency, the surgeon needs to change his key value. Then, he can search that patient's index and call Increase key function to manage his patient database.



[3]

[1] Bae Hunsang(2020. 3. 31). priority_queue.c [C code] https://github.com/baehunsang/Algorithm---/blob/master/priority_queue.c

[2] Compare part of heapify function. Compare function is needed to manage other data type except integer.

[3] Shafekul Abid(2018. 11. 26). Priority Queue | Priority Based Queues | Data Structure | CS School https://csschoolonline.com/wp-content/uploads/2018/11/real-life-priority-queue.png