CMSC 123: Data Structures

1st Semester AY 2020-2021

Prepared by: CCS Templado & KBP Pelaez

Exercise 06: Heap ADT (Insert Function)

Heap ADT

Even though the heap is another binary tree data structure, for this exercise, you must implement heaps using arrays.

To create a `HEAP ADT` that will hold integers, we define the following structure (defined in `heap.h`):

```
typedef struct heap_tag{
    int *heap;    //pointer to an array of integers
    int size;     //size of the heap
    int maxSize;  //max allowed size of heap
    int type;   //type of the heap (i.e., 0-min or 1-max)
} HEAP;
```

Tasks

Implement and test the following functions (also listed in `heap.h`):

1. `HEAP* createHeap(int maxSize, int type);` - a function returns a pointer to a heap whose maximum size is `maxSize` and whose type is defined by `type`.
2. `int isFull(HEAP *H);` - a function that returns 1 if the heap is full, otherwise, 0.
3. `int isEmpty(HEAP *H);` - a function that returns 1 if the heap is empty, otherwise, 0.
4. `void clear(HEAP *H);` - a function that deletes all the contents of the heap.
5. `void insert(HEAP *H, int key);` - a function that properly inserts `key` to the heap.

The implementation for `printHeap` is already in `heap.c`.

Make sure to test your program using a shell file. Format is as follows:

1. Line 1 should contain either 0 or 1. 0 for MINHEAP and 1 for MAXHEAP.
2. Succeeding lines must contain one of the following commands:
   - `+ i` - inserts `i` to the heap
   - `p` - prints the heap
   - `E` - checks if the heap is empty
   - `F` - checks if the heap is full
   - `C` - clears the contents of the current heap
3. The last line in the file must contain the `Q` command for the program to terminate.

Submission

Submit your `heap.c` to Google Classroom.

Questions?

If you have any questions, approach your lab instructor.