Lab #13

CS-2050

May 3, 2024

1 Requirements

It is *highly suggested* that you make use of helper functions to complete this lab.

In this lab, you are tasked with implementing functions for the purpose of creating and maintaining a linked heap. Functions worth 0% won't be considered part of your grade, but you **must** implement them in order for your submission to be graded.

Note the complexity requirement specified on each function. If your implementation of a function does not meet the complexity requirement specified, you cannot get credit for that function.

1.1 makeHeap

```
// O(1)

Heap * makeHeap(void)
```

Info: This function creates and returns an empty **MAX heap**. If creating the heap was successful, it returns a pointer to the heap, otherwise it returns NULL.

1.2 compareHeap

```
// O(1)

int compareHeap(void *a, void *b)
```

Info: **you MUST implement this function in your main.c**, we will provide our own comparison function during grading. You **must not** submit this function with your implementation code. Your code must assume that this function is provided elsewhere (like malloc() or printf()).

This function takes two data, and compares them. It returns a positive number if (a > b), zero if (a == b), and a negative number if (a < b). You can find an example implementation of a user-provided comparison function in the lab 12 solution on Canvas.

1.3 insertHeap

```
// O(log(n))

int insertHeap(Heap *heap, void *data)
```

Info: This function takes a heap, and inserts the given data onto the heap. It returns 1 on success, or 0 on failure. You are required to use randomization to maintain balance during insertion. You may assume that NULL is never inserted.

1.4 deleteMax

```
// O(log(n))
```

```
void * deleteMax(Heap *heap)
```

Info: This function takes a heap, removes the **MAX element** from the heap, and returns the removed element. It returns NULL if the heap is empty.

1.5 destroy

```
// O(n)
```

```
void destroy(Heap *heap)
```

Info: This function takes a heap, and frees all memory allocated to it.

Grading: 100 %

1. Write required *makeHeap* function
   * 0 %
2. Write required *compareHeap* function
   * 0 %
3. Write required *insertHeap* function
   * 30 %
4. Write required *deleteMax* function
   * 50 %
5. Write required *destroy* function
   * 20 %

Notice:

1. All of your lab submissions **must** include documentation to receive full points.
2. All of your lab submissions must compile under GCC using the $-Wall$, $-Werror$, and $-Wpedantic$ flags to be considered for a grade.
3. You are expected to provide proper documentation in every lab submission, in the form of code comments. For an example of proper lab documentation and a clear description of our expectations, see the lab intro PDF.