lab 24 Quad Trees

Instructions: In this lab implement these basics of a QuadTree.

Implement the following class:

```cpp
#ifndef QUADTREE_H__
#define QUADTREE_H__

#include <stdlib.h>
#include <vector>
#include <list>
#include <set>
#include <queue>

class QuadTree {
    private:
        /* Class to begin filling out...*/
    public:
        /* Initialize an empty quadtree. */
        QuadTree(float width, float height);

        /* add a point to the quadtree. */
        bool add(float x, float y);

        /* remove a point from the QuadTree.
         * Remember to remove empty QuadTree nodes, or your tree will
         * use up too much memory when doing the add/remove test!
         */
        bool remove(float x, float y);

        /* returns if the quad tree has a point (x, y) */
        bool contains(float x, float y);

        /* return the number of points in the box (sx, sy) -> (ex, ey)
         * You may assume that sx < ex and sy < ey!
         */
        int countInRange(float sx, float sy, float ex, float ey);

        void print();
};

#include "quadtree.cpp"

#endif
```

Write some test cases:

Create some test cases, using cxxtestgen, that you believe would cover all aspects of your code.

Memory Management:
Now that are using new, we must ensure that there is a corresponding delete to free the memory. Ensure there are no memory leaks in your code! Please run Valgrind on your tests to ensure no memory leaks!

How to turn in:
Turn in via GitHub. Ensure the file(s) are in your directory and then:

- $ git add <files>
- $ git commit
- $ git push

Due Date: December 04, 2019 2359

Teamwork: No teamwork, your work must be your own.