

# CMPT 225 Lab 7: Heaps

---

We will finish an implementation of a heap of strings, and test the implementation. Your heap should implement a priority queue (where the highest priority item is always removed first). Your heap should be a min heap, so the root should always contain the string that comes first in alphabetical order. (By "alphabetical order," we mean more precisely *ASCII lexicographic order*.)

Start by downloading [the zipfile](#). It contains:

- heap.cpp, which you should fill in
  - heap.h, a header file that is complete
  - heap\_test.cpp, a test driver that you should not modify
  - heap\_debug.cpp, a debug driver that you can modify to debug your heap
  - A makefile.
  - test.py and a set of \*.in and \*.gt input and ground truth files
- 

## Notes

### Implementation

You should implement your heap as discussed in lecture. The remove method (and *bubbleDown* (downheap)) has been provided. Please write the insert method. Write a separate private *bubbleUp* (upheap) method that should be called by your insert method.

### Comparing Strings

We are using the C++ strings defined in `<string>`, `std::string` (using namespace `std` you can omit the `std::`). If you want to compare two strings you can use the operators `>`, `<`, `==`, since these are properly overloaded for the string class.

---

## Testing

Test your Heap class by running the python script test.py.

```
uname@hostname: ~$ ./test.py
```

**Please run this for a TA successfully (3 tests passed) to receive your marks for this lab.**

---

## Debugging

Please modify the driver heap\_debug.cpp if you wish to add debugging calls. Final versions of Heap methods should not produce any output to cout, since cout's output is used for correctness checks

by test.py. If your code fails a test case, you can try running it individually, and comparing it to the corresponding ground truth file. E.g. to run test case 1 and examine its ground truth:

```
uname@hostname: ~$ ./heap_test < 1.in  
uname@hostname: ~$ more 1.gt
```