

## Heaps

### 1. Binary Heap

- a. Write a program `bheap.cpp` that implements the Binary Heap data structure. You may use the textbook implementation. You may assume that all keys are integers and are distinct.
- b. Extend your program with an interpreter (in the style of `p2.cpp` from Program 2) that executes a sequence of operations from an input file on an initially empty heap. There are three types of operations:
  - `insert k` inserts an integer  $k$  into the heap.
  - `deletemin` deletes the minimum integer from the heap.
  - `print` prints the sequence of integers in the array implementation of the heap. For example, you should print the following sequence for the heap in Figure 6.5 of the textbook: 13 21 16 24 31 19 68 65 26 32Test your program using the input file `bheap.txt`, which simulates the operations in Section 6.3.3 of the textbook.
- c. Extend your program such that it performs the following tasks when no input file is given:
  - i. Read an integer from the standard input. Let this integer be  $n$ .
  - ii. Insert the  $n$  keys  $n, n - 1, \dots, 1$ , in decreasing order, into an initially empty heap.
  - iii. Print the height of the heap, then exit. (Hint: you can calculate the height directly from  $n$ .)
- d. Use a stop-watch to measure the running times  $T(n)$  of your program in seconds for increasing values of  $n = 2^k$  that are powers of two. For the values  $T(n)$  that are at least 10 seconds and at most 600 seconds, plot  $\log T(n)$  versus  $\log n$ , then fit it with a straight line. (You may plot either by hand or by Excel.) It is mentioned on page 218 of the textbook that the average running time of an insert operation on a binary heap is a constant. Does your plot support this claim?

### 2. Leftist Heap

- a. Write a program `lheap.cpp` that implements the Leftist Heap data structure. You may use the textbook implementation. You may assume that all keys are integers and are distinct.
- b. Extend your program with an interpreter (in the style of `p2.cpp` from Program 2) that executes a sequence of operations from an input file on an initially empty heap. There are three types of operations:
  - `insert k` inserts an integer  $k$  into the heap.
  - `deletemin` deletes the minimum integer from the heap.
  - `print` prints the sequence of integers along the right-most path (from the root to a node with a null right child) of the heap. For example, you should print the following sequence for the heap in Figure 6.24 of the textbook: 3 10 14

- Test your program using the input file lheap.txt.
- c. Extend your program such that it performs the following tasks when no input file is given:
    - i. Read an integer from the standard input. Let this integer be  $n$ .
    - ii. Insert the  $n$  keys  $1, 2, \dots, n$ , in increasing order, into an initially empty heap.
    - iii. Print the null path length of the root of the heap, then exit.
  - d. Use a stop-watch to measure the running times  $T(n)$  of your program in seconds for increasing values of  $n = 2^k$  that are powers of two. For the values  $T(n)$  that are at least 10 seconds and at most 600 seconds, plot  $\log T(n)$  versus  $\log n$ . Can you fit it with a straight line? (You may plot either by hand or by Excel.) Is the average running time of an insert operation on a leftist heap a constant?

Submit two C++ files bheap.cpp, lheap.cpp, and a document that describes the result your experiments on  $T(n)$  in a zip file to eagle.