Data Structures and Algorithms: Homework #5

Due on May 26, 2015 at 16:20

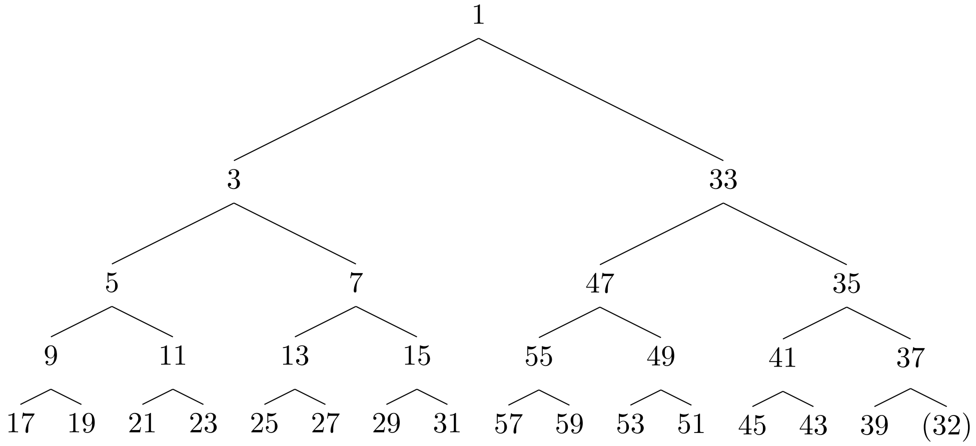
Instructors Hsuan-Tien Lin, Roger Jang

Tim Liou (b03902028)

5.1 Heap and Hash

(1) Complete Exercise R-8.24 of the textbook.

Consider a min heap. The insertion of an entry with key 32 would cause up-head bubbling to proceed all the way up to a child of the root in this heap.



(2) Complete Exercise C-8.4 of the textbook.

Combine each key we want to insert to the stack with a priority variable. These combinations would be insert to the MaxPriorityQueue which compare elements by the priority variable. Note that the basic idea of this method is the fact that every new element is pushed in stack with a priority higher than the current one. Then, MaxPriorityQueue will behave in the LIFO way, which is what we want.

```
class Stack {
    class Element { int priority, Key element; };
    MaxPriorityQueue<Element> MaxPQ;
    int top_priority = 0;

    void push(Key key) { MaxPQ.insert(Element(top_priority++, key)); }
    void pop() { top_priority--; MaxPQ.removeMax() }
    const Key& top() { return MaxPQ.max(); }
    int size() { return top_priority; }
    bool empty() { return (top_priority == 0); }
};
```

(3) Complete Exercise C-8.14 of the textbook.

```
function FindLeqElemInHeap(heap, key)
    while heap.top() $\leq$ key do
        insert heap.top() into the output list, and do heap.removeMin()
    end while
    return the output list
end function
```

Algorithm 1: Compute all the entries in a heap with a key less than or equal to the value

(4) Hash function is everywhere. Use any search engine to study the term MinHash Explain to the TAs what it is and why it is useful. Also, cite the website that you learn the term from.

====== Pending ======

(5) Describe an algorithm to find out the position that the two strings differ efficiently. Briefly discuss and justify the time complexity of your algorithm.

====== Pending ======

(6) Construct a perfect hash function that is efficiently computable for the following 32 standard keywords in C. You need to explain why the hash function is perfect and why it is efficiently computable to get the full bonus.

====== Pending ======

5.2 Distributed System

(1) Finish/rewrite the BinomialHeap class, and describe how you test whether the data structure is correct.

====== Pending ======

(2) Implement the system for three kinds of commands below.

====== Pending (code part) ======