CMPS 6610 Algorithms – Fall 16

10/25/16

5. Homework

Due **11/1/16** at the beginning of class

1. **Aggregate Analysis (9 points)**

   Consider a sequence of n operations on a data structure, in which the cost c_i of the i-th operation is defined as $c_i = i^2$ if i is a power of 2, and $c_i = 1$ otherwise.

   (a) (1 point) What is the worst-case runtime of a single operation?

   (b) (4 points) Use aggregate analysis to determine the amortized cost per operation.

   (c) (4 points) Use the accounting method to determine the amortized cost per operation.

2. **Stack Backups (4 points)**
   A sequence of stack operations (push or pop) is performed on a stack whose size never exceeds k. After every k operations, a copy of the entire stack is made for backup purposes. Show that the cost of n stack operations, including copying the stack, is $O(n)$ by using the accounting method to assign suitable amortized costs to the stack operations.

3. **Amortizing Heaps (5 points)**
   Consider a binary max-heap data structure with n elements that supports the operations INSERT and EXTRACT-MAX in $O(\log n)$ worst-case time. Give a potential function Φ such that the amortized cost of INSERT is $O(\log n)$ and the amortized cost of EXTRACT-MAX is $O(1)$, and show that it works.
   (Hint: Let n_i be the number of elements in the heap after i operations. Consider a function based on $n_i \log n_i$.)

4. **Adjacency Matrix (4 points)**
   What are the runtimes of Depth-First Search and of Breadth-First Search if the graph is given in an adjacency matrix? Justif your answers.

5. **Finding a Cycle (4 points)**
   Give an algorithm that decides whether an undirected graph $G = (V, E)$ contains a cycle. Your algorithm should run in linear time, and with the right arguments you should be able to make it run in $O(|V|)$ time only.

6. **Stack from Queues (4 points)**
   Show how to implement a stack using two queues. Analyze the runtime of the stack operations.