CS 161 - Design and Analysis of Algorithms (Summer 2012) — Homework 1
Stanford University — Due: Friday, July 6, 2012 at 2:15PM

Problem 1 (15 Points). Order the following functions by big Oh. You should list the functions by increasing Big Oh, showing which functions are Big Theta of each other. For each two adjacent functions in your list, please provide a justification for the ordering (i.e. show that $f_i(n) \in O(f_j(n))$, and why $f_i(n)$ is or isn't $\Theta(f_j(n))$).

$$
\begin{array}{lll}
f_1(n) = 100\sqrt{n} & f_2(n) = 2\log(10n) & f_3(n) = 3^n \\
f_4(n) = 2^n & f_5(n) = n^2 & f_6(n) = 6\log n \\
f_7(n) = n^{1/10000} & f_8(n) = \log(n^{10}) & f_9(n) = n2^n \\
f_{10}(n) = (\log n)^{\log n} & f_{11}(n) = 10n & f_{12}(n) = n^{\log n} \\
f_{13}(n) = (\log n)^{10000} & f_{14}(n) = 20n\log n & f_{15}(n) = n + 8\log n
\end{array}
$$

Problem 2 (15 Points).

(a) Suppose $f(n) \notin O(1)$. Show that $f(n)^2 \notin O(f(n))$. Conclude that there is no asymptotically largest function.*[Hint: Show the contrapositive, that if $f(n)^2 \in O(f(n))$, then $f(n) \in O(1)$]*

(b) Suppose $f(n) \notin O(1)$. Show that $2^{f(n)} \notin O(f(n))$.

Problem 3 (5 Points). Show how to use an array to implement a queue with all operations being (amortized) constant time.

Problem 4 (10 Points).

(a) Show how to implement a stack using two queues. Analyze the running time of the stack operations.

(b) Show how to implement a queue using two stacks. Analyze the running time of the queue operations.

Problem 5 (15 Points). We want hash functions to look like random functions, so why can't we just use a truly random function? Let $\mathcal{X}$ be a set with N elements, and $\mathcal{Y}$ a set with M elements.

(a) Show that the number of functions from $\mathcal{X}$ to $\mathcal{Y}$ is M^N.

(b) Suppose $\mathcal{X}$ is the set of n-bit sequences, and $\mathcal{Y}$ is the set of m-bit sequences. The random hash function needs to be written down somewhere. Show that the hash function can be represented using $m2^n$ bits.

(c) Show that this representation is tight: that any representation of all functions from $\mathcal{X}$ into $\mathcal{Y}$ must have at least one function that requires $m2^n$ bits. Assuming the number of subatomic particles in the visible universe is bounded by 10^{100}, and each subatomic particle can represent 1 bit, and that we are interested in hashing strings into single bits ($m = 1$), what length of sequences can we possibly hope to hash using truly random functions?

Problem 6 (15 Points). Suppose we want to create a hash table whose keys are student names (represented as a sequence of lower-case letters), and whose values are their GPAs. Let the size of the hash table be N. Explain why the following candidate functions are poor choices:

(a) Choose random integers A and B in $\{0, ..., N-1\}$. Define $h_1(s)$ as follows: decompose the string s into characters $s_1, ..., s_n$, and interpret these characters as integers, where a space is 0, "a" $= 1$, "b" $= 2$, etc. Then $h_1(s) = A(s_1 + ... + s_n) + B \mod N$.

(b) Using the same interpretation of characters as integers, $h_2(s) = s_n + 31s_{n-1} + 31^2 s_{n-2} + ... + 31^{n-1}s_1 \mod N$.

(c) $h_3(s) = t \mod N$ where t is the time, rounded to the nearest millisecond, when the function h_3 is called.

Problem 7 (10 Points). Prove the time bounds for a d-ary heap. That is, show that:

(a) `deletemin` operations take $O\left(\frac{d \log |V|}{\log d}\right)$.

(b) `insert` and `decreasekey` operations take $O\left(\frac{\log |V|}{\log d}\right)$.

Problem 8 (15 Points). Show how to implement the following features on a balanced BST with n nodes:

(a) `range`(r, a, b) takes as input the root r of the tree and two values a and b with $a \leq b$, and returns the sorted, possibly empty list of values between a (inclusive) and b (exclusive). If k values are output, your algorithm should run in time $O(k + \log n)$.

(b) `numInRage`(r, a, b) counts the number of values between a (inclusive) and b (exclusive). Your algorithm should run in time $O(\log n)$, independent of the actual number of values in the range. This will require storing additional data in the tree. Explain how to modify insertions, deletions, and rotations to maintain the new data.

Total points: 100