Data Structures - Assignment no. 8, June 20, 2007

Remarks:

- **Write both your name and your ID number very clearly on the top of the exercise. Write your exercises in pen, or in clearly visible pencil. Please write *very* clearly.**
- **Recall that 80% of the theoretical exercises must be submitted. The exercises can and must be worked on and submitted alone.**
- **Give correctness and complexity proofs for every algorithm you write.**
- **For every question where you are required to write pseudo-code, also explain your solution in words.**

1. Find the order of growth for the following recursively given functions, using the Master Method.

   (a) $T(n) = 4T(n/2) + n$

   (b) $T(n) = 4T(n/2) + n^2$

   (c) $T(n) = 4T(n/2) + n^3$

   (d) Explain why $T(n) = 2T(n/2) + nlog^2 n$ cannot be solved using the Mater Method.

2. Find the order of growth for the following recursively given functions. Explain your answer.

   (a) $T(n) = T(n-a) + T(a) + n$ where $a \geq 1$ is constant.

   (b) $T(n) = T(cn) + T((1-c)n) + n$ where $0 < c < 1$ is a constant.

   (c) $T(n) = 2T(n/2) + nlog^4 n$ (Hint: The approach to solving this is similar to the approach to solving $T(n) = 2T(n/2) + n$).

3. [1] Consider a hash table of size $m = 1000$ and the hash function

   $$h(k) = \lfloor m \cdot (k\theta \ mod \ 1) \rfloor$$

   for $\theta = (\sqrt{5} - 1)/2$. Here, $x \mod 1$ is the fractional part of the real number x, e.g. 15.1345 mod 1 = 0.1345. Compute the locations to which the keys $61, 62, 63, 64, 65$ are mapped.

4. [2] A family $\mathcal{H}$ of functions from U to $\{0, 1, \ldots, m-1\}$ is called *100-weakly universal* if for all $x, y \in U$ such that $x \neq y$ it holds that

   $$Pr_{h \leftarrow \mathcal{H}}[h(x) = h(y)] \leq \frac{100}{m} .$$

   Let $U = \{0, 1, \ldots, u-1\}$. Define f_a to be the function $f_a(x) = a \cdot x \mod m$. Show that $\mathcal{H} = \{f_a | 0 < a < m\}$ is not a 100-weakly universal family. (Hint: take u to be very large, compared to m).

[1] borrowed from LSE exercise on hash tables

[2] borrowed from course at Bar-Ilan

5. Show how to modify the Hash Table data structure, such that all operations (find, insert and delete) take $O(\log n)$ time in the worst-case. All operations should still take $O(1)$ time in expectation. This question refers to the hash table data structure that you learned in class, using universal hashing and chaining. This does not refer to perfect hashing.

6. [3] A closed hash table of size $2n$ is used to store n items. (Closed hash is sometimes also called "open addressing").

   (a) Assuming uniform hashing (i.e. each probe is done at a random location), show that for $i = 1, 2, \dots, n$, the probability that the i^{th} insertion requires strictly more than k probes is at most 2^{-k}.

   (b) Show that for $i = 1, 2, \dots, n$, the probability that the i^{th} insertion requires more than $2 \log n$ probes is at most $1/n^2$.

   (c) Let the random variable X_i denote the number of probes required by the i^{th} insertion. You have shown in part (b) that $Pr(X_i > 2 \log n) \leq 1/n^2$. Let the random variable $X = \max_{1 \leq i \leq n} X_i$ denote the maximum number of probes required by any of the n insertions. Use the union-bound to show that $Pr(X > 2 \log n) \leq 1/n$.

   (d) Use Markov's inequality this fact to show that the expected length of the longest probe sequence is $E[X] = O(\log n)$.

[3] taken from CLRS