CSc445 Homework #3. Due: 6/12/2016 Midnight

Instructions.

1. You could submit a pdf of the homework, either printed or hand-written and scanned, as long as it is easily readable.

2. If your solution is not clearly written, it might not be graded.

3. Prove the correctness of your answer. A correct answer without a proof might not be accepted.

4. If you have discussed the solution with other students, mention their names clearly on the homework. These discussions are not forbidden and are actually encouraged. However, you must write your whole solution yourself.

5. All questions have same weight.

6. You could refer to a data structure studied in class, and just mention briefly their guarantees. For example “It is known that a Red-Black tree could support the insert/delete/find operations on a set of $n$ elements in time $O(\log n)$.”

7. If your answer uses one of the data structures or algorithms that were studied in class, you could refer to it without having to repeat details studied in class. If your answer requires only modifications of one of the algorithm, it is enough to mention the required modifications, and what’s the effect (if any) on the running time and on other operations that the algorithm performs.

8. In general, a complete answer should contain the following parts:

(a) High level description of the data structures (if needed). E.g. We use a binary balanced search tree. Each node contains, a key and pointers to its children. We augment the tree so each node also contains a field...

(b) High level description of the algorithms

(c) Proof of correctness (why your algorithm provides what is required).

(d) A claim about the running time, and a proof showing this claim.
1. Assume a hash table $T[0..15]$ (that is, $m = 16$), and a open addressing hashing where

$$h(x,i) = \{x + i \cdot (x \mod 10)\} \mod m.$$ 

Assume you start with an empty table. Show an example of a set of three distinct keys $\{k_1, k_2, k_3\}$ such that

(a) You could not insert all of them into the table. That is, calling $\text{insert}(k_1)$, followed by $\text{insert}(k_2)$, followed by $\text{insert}(k_3)$ would report that the last operation is unsuccessful. And in addition,

(b) $k_j \mod 10 > 0$ for $j = 1, 2, 3$.

2. Under the same assumptions as the previous question, but now pick $m = 17$ (rather than $m = 16$). Prove that there is no such set of 3 different keys.

3. Under the same assumptions as the previous question, but now pick $m$ to be any prime $\geq 17$. Prove that there is no such set of 3 different keys. You could use the following known result. Assume $m$ is a prime. Let $a, r, t$ be three integers, all $\geq 1$ and $\leq m - 1$. Assume $r \neq t$. Then

$$(ar) \mod m \neq (at) \mod m.$$ 

4. Under the same notations as the slides. Assume $h(x)$ is any hash function, mapping keys from a universe $U$ onto a table $T[0..m-1]$. Show that if $|U| \geq m^2$ then there is a set $K \subset U$ such that all keys of $K$ are mapped by $h(x)$ into the same slot of $T$ and $|K| \geq m$. That is, every pair of keys from $K$ creates a collision.

5. Discuss pros and cons of Open Addressing Hashing (where collisions are resolved using double probing) vs. Chain Hashing.

6. Explain how you would use hash functions to find if your computer contains two identical image files (possibly under different names). Give a pseudocode of your solution. Specify which and how your hash functions are used. Do not use values provided by the file system. Your algorithm should be as efficient (in space and running time) as possible.

Assume images are stored as raw data. That is, images are given as matrix of pixels, and for each pixel, we are given the RGB values, as numbers between 0 and 255.

7. Repeat the same question, but now assume that two images $M$ and $M'$ are considered identical if there is some constant $\beta$ such that each pixel of $M$ is obtained the corresponding pixel of $M'$ after multiplying it by $\beta$. That is, $M[i,j] = \beta \cdot M'[i,j]$ for every $i, j$.

Think about $\beta$ for example as changing the intensity of the image. Note that $\beta$ is not known to you.

Ignore the effect of numeric computation errors caused by truncation. That is, on you computer, (miracle), $\beta(\frac{k}{5}) = k$, for any integer $k$. For example, $3 \left(\frac{1}{5}\right) = 1$ and not 0.99999.

You could assume that the blue and red values always equal to zero. So $M[i,j]$ is the value of the green level at pixel $[i,j]$. 


8. Repeat Question 6, but this time you could use values provided by the File System/Operating System (such as MD5). Could you use the MD5 value of the file as an index to a hash table?

9. Consider the process of inserting \( m \) keys into a hash table \( T[0..m-1] \), where \( m \) is a prime, and we use double addressing. The hash function we use is

\[
h(k, i) = (k + i) \mod m
\]

Give an example of \( m \) keys \( k_1, k_2 \ldots k_m \), such that the sequence of operations

\[
\text{insert}(k_1) \\
\text{insert}(k_2) \\
\vdots \\
\text{insert}(k_m)
\]

Takes \( \Omega(m^2) \) time.

10. Consider a open addressing function \( h(k, i) = (h_1(k) + i) \mod m \) that maps keys into a hash table \( T[0..m-1] \). Here \( h_1(k) = (k \cdot B) \mod m \), and \( B \) is any fixed integer.

You have inserted the set of keys \( j \cdot m \) (for \( j = 0, 1, 2 \ldots \lceil \frac{m}{2} \rceil \)). Next you are about to insert another single key \( k \).

Assume that the key \( k \) could be mapped into each one of the \( m \) slots with equal probability. What is the expected time that this operation would take?

If you are not familiar with computing expectation, just discuss the possible cases for the time needed to insert \( k \) (the answer is different, depending on what is the value of \( h_1(k) \)).