Worth: 11% **Due:** Wednesday March 10

For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing.

1. Let U be a universe that contains character-strings. Assuming that the characters have ASCII values in the set $\{0, 1, \ldots, 127\}$ we may use the radix-128 representation in order to convert the strings into integers: for a string with characters of (ASCII) values $\sigma_0, \sigma_1, \ldots, \sigma_r$ we use the integer value $k = \sum_{i=0}^{r} \sigma_i 128^i$.
Now suppose we use the hash function $h(k) = k \bmod m$, where $m = 127$. Show that if y is a permuted version of string x (say $x =$stop, and $y =$spot), then we have a collision between x and y.

2. Consider two hash tables, T_1 and T_2, with the same number of buckets. With T_1, we use chaining to resolve collisions, where each chain is a doubly-linked list and insertions are done at the front of the list. With T_2, we use linear probing to resolve collisions, and deletions are done by replacing the item with a special "deleted" item. For both tables, we use the same hash function.
Now, Let $S = S_1, S_2, ..., S_n$ be a sequence of INSERT, DELETE and SEARCH operations, and suppose that we perform S on both T_1 and T_2. For $1 \leqslant i \leqslant n$, let $C_{1,i}$ be the number of item comparisons made when performing S_i on T_1, and let $C_{2,i}$ be the number of item comparisons made when performing S_i on T_2.

a. Give an example of a sequence S in which $C_{1,i} > C_{2,i}$ for some $1 \leqslant i \leqslant n$ where S_i is a SEARCH operation.

b. Let S be any sequence of INSERT, DELETE and SEARCH. Suppose we perform S on both T_1 and T_2, and then perform a SEARCH operation where each item in the table is *equally likely* to be searched for. Let E_1 be the expected number of item comparisons made when performing the SEARCH operation on T_1, and let E_2 be the expected number of item comparisons made when performing the SEARCH operation on T_2. Prove that $E_1 \leqslant E_2$.

3. Suppose that we are using a hash table with m buckets, where $m > 1$. Recall that in open addressing we store an element in the first empty bucket among $A_0(k), A_1(k), A_2(k), \ldots$ that is empty, where $A_i(k) \in \{0, 1, 2, \ldots, m-1\}$ is defined by the probing mechanism. A version of quadratic probing, a little different than the one we saw in class, uses the following function.

$$A_i(k) = (h(k) + i^2 + i) \bmod m$$

where $h(k)$ is some hash function.
Prove that the probe sequence of this method will always include at most $(m + 1)/2$ distinct buckets. Why is this a problem?