CS 261 — Spring 2025 — Midterm

Name:

UCI email:

Do not open this exam until told to start by the instructor.

This is a closed book, closed note exam. Calculators or other electronic devices are not allowed.

Please write your answers ONLY on the front side of each page.
Answers written elsewhere will not be scanned and will not graded.

You may use the back sides of the pages as scratch paper.
Do not unstaple the pages.

1. (15 points) Consider a (very inefficient) priority queue that stores its values in unsorted order in an array. When it inserts a new value, it just adds it to the end of the array, in constant actual time. To perform a delete-min operation, it scans the whole array to find the minimum priority, and then moves the element from the end of the array into that position, taking actual time $O(n)$ when there are n elements in the priority queue. But instead of actual time we are going to analyze this data structure using amortized time, with the potential function $\Phi = n(n+1)/2$, so that adding one element increases the potential function by n (the size of the priority queue after the insertion) and removing an element decreases the potential function by the same amount.

   (a) For this data structure and this potential function, what is the amortized time for an insert operation? Express your answer using O-notation.

   (b) For this data structure and this potential function, what is the amortized time for a delete-min operation? Express your answer using O-notation.

2. (15 points) Suppose that we have a hash table whose size is a (non-random) prime number p, that the keys we are hashing are integers, and that we define our hash function by choosing random numbers a and b modulo p and hashing each number x to the hash table cell with index $ax + b$ modulo p. Demonstrate that this is not 2-independent by finding two values x and y that are always mapped to the same cell as each other. (Hints: your choices of x and y can depend on p, and can be larger than p. It is these assumptions that make this hash function different from the 2-independent function in the lecture notes.)

3. (15 points) Recall that the load factor α of a hash table is its number of key–value pairs divided by its number of cells. The hashing algorithms that we have seen in class work best when α is small, but they have different requirements on how small α needs to be. For each of the following questions, you may assume that the hash function is random.

   - For which values of α will linear probing give $O(1)$ expected time per operation?

   - For which values of α will hash chaining give $O(1)$ expected time per operation?

   - For which values of α will cuckoo hashing successfully insert all elements, with high probability? (Use the same version of cuckoo hashing as described in class: one key per cell, and two hash locations per key).

4. (15 points) Suppose that X, Y, and Z are sets represented as binary numbers. Write down an expression using any of the bitwise binary operations & (and), | (or), ~ (not), or ^ (exclusive or) for the set of elements that belong to at least two of the sets X, Y, and Z.

5. (15 points) For a Bloom filter with n elements, $2n$ cells, and each element hashed to two random cells, write a formula for the expected number of cells with value zero. (Hint: first calculate the probability that a single cell has value zero and then use linearity of expectation. Your formula should be exact; you do not need to use the approximation $(1-1/x)^x \approx 1/e$.)

6. (15 points) Recall that the Boyer–Moore majority sketch maintains two variables, m, an element that it thinks might be a majority element, and c, a count of occurrences of that element.

   (a) If this data structure is applied to the six-element sequence of elements A, B, B, C, C, C, what will be the values of m and c at the end of the sequence?

   (b) Find a different ordering for the same six elements so that, using your reordered sequence instead of the sequence above, the value of m at the end of the sequence is equal to A.

7. (15 points) Suppose that we have a stream of element values of length n, that there are d different values in the stream, and that element x repeats $r > 0$ times in the stream.

   (a) If we use reservoir sampling to maintain a sample of k values from the stream, what is the probability that x is one of the sampled values? (You may assume that k is small enough that there are more than k choices for what to sample.)

   (b) If we use MinHash to maintain a sample of k values from the stream, using a random hash function, what is the probability that x is one of the sampled values? (Again, you may assume that k is small enough that there are more than k choices for what to sample.)

8. (15 points) Suppose that, before a delete-min operation, a Fibonacci heap has the following structure: the minimum-priority tree root has three children, whose degrees are 0, 1, and 2. Additionally, there are two more trees in the forest, one with a root of degree 0 (a one-vertex tree) and one with a root of degree 2. After the delete-min operation, how many trees will the heap have, and what will their degrees be?