CS 584 Algorithms: Design and Analysis
Fall term 2019

Practice Midterm Exam

These problems are sample problems for the midterm exam, so you may expect similar problems in the midterm. Do not hand in your solutions. Solutions will be discussed (and posted on the web) in class. The midterm exam is a closed book exam, but you may use one sheet of paper (written on both sides) with notes on it.

1. Use the Master Theorem to find the asymptotic solutions for the following recurrences: (a) \( T(n) = 7T\left(\frac{n}{2}\right) + n^2 \), (b) \( T(n) = T\left(\frac{n}{2}\right) + 1 \), (c) \( T(n) = 4T\left(\frac{n}{2}\right) + n^3 \). (10 points)

2. Use indicator random variables to find the expected value of the number of fixed elements (elements left in the same position) in a random permutation of \( n \) elements. (15 points)

3. We have two input arrays, an array \( A \) with \( m \) elements and an array \( B \) with \( n \) elements, where \( m \leq n \). There may be duplicate elements. We want to decide if every element of \( B \) is an element of \( A \). Describe an algorithm to solve this problem in \( O(n \log m) \) worst-case time. (15 points)

4. Assume that you want to sort an array of \( n \) numbers, each of which is a member of the set \( \{0, 1, 2, 3, 4\} \). A sample input for \( n = 6 \) is \( (3, 1, 0, 3, 4, 3) \). Describe an optimal algorithm to solve this problem and prove that it is optimal; that is give a lower bound on the worst-case running time that has the same order of magnitude as the worst-case running time of your algorithm. (15 points)

5. We have an input array \( A \) with \( n \) \( (n \geq 2) \) numbers.

   (a) Describe a \( O(n) \) worst-case time algorithm to find two elements \( x, y \in A \) such that \( |x - y| \geq |u - v| \) for all \( u, v \in A \).

   (b) Describe a \( O(n \log n) \) worst-case time algorithm to find two elements \( x, y \in A \) such that \( |x - y| \leq |u - v| \) for all \( u, v \in A \).
6. An array $A[1..n]$ of $n$ distinct numbers is called \textit{unimodal} if there is a unique \textit{mode} $j$, such that $A[i] > A[i+1]$ for $j \leq i < n$ and $A[i-1] < A[i]$ for $1 < i \leq j$.

(a) Assume that you have an algorithm that tests whether $A$ is unimodal and if it is then finds the mode. Use an adversary argument to show that your algorithm must have a worst-case time complexity $\Omega(n)$.

(b) Assuming that $A$ is unimodal, give an algorithm for finding the mode for which the worst-case time complexity is $O(\log n)$. Isn’t this contradictory with part (a)?

(15 points)

7. In a variant of open-address hashing we insert the elements in pairs $(k_1, k_2)$ into a hash table, that is we insert $k_1$ into the first empty slot in the probe sequence, then we start from scratch and insert $k_2$ into the first empty slot again. What is the expected number of probes needed to insert a pair of keys into a hash table with load factor $\alpha = n/m$, assuming uniform hashing? (15 points)