

Some CPSC 259 Sample Final Exam Questions (Part 4)

Some of these have appeared on exams for related courses. I've changed some questions to reflect CPSC 259 material (e.g., translated the code to C, excluded material we don't cover, included material we do cover, and changed the marks a bit to reflect probable CPSC 259 marking practices).

1. {4 marks} What is the worst-case running time for inserting n items into an initially empty hash table, where collisions are resolved by chaining? What if each sequence (chain) is stored in sorted order?
2. {4 marks} Suppose that each row of an $n \times n$ array A consists of 1's and 0's such that, in any row of A , all the 0's come before any 1's in that row. Assuming that A is already in memory, describe a function running in $O(n \lg n)$ time (*not* $O(n^2)$ time) for counting the number of 1's in A .
3. {4 marks} Let A be a collection of objects. Describe (in words) an efficient function for converting A into a set. That is, remove all duplicates from A . What is the running time of this method?
4. {6 marks} Suppose we want to sort an array of size n that contains items that are either 0 or 1.
 - a) Use Big-O notation to describe the asymptotic worst-case number of comparisons made by Insertion Sort to sort such an array. (No proof is necessary.)
 - b) Describe a worst-case input for (a), for the number of entries n .
 - c) Give Big-O notation to describe the asymptotic worst-case scenario for the number of comparisons made by Quicksort to sort such an array. (Again, no proof is necessary.)
 - d) Describe a worst-case input for (c), for the number of entries n .
5. {4 marks} In matrix multiplication, the product of two $n \times n$ matrices A and B is the one $n \times n$ matrix C whose i,j -th entry is computed as follows:

$$c_{i,j} = \sum_{k=1 \text{ to } n} a_{i,k} * b_{k,j}$$

If we wrote an algorithm that multiplied matrices in this way, how many scalar multiplications (i.e., “*” operations) would the algorithm perform when multiplying one $n \times n$ matrix by another $n \times n$ matrix? Use Big-O notation. (You do not need to find witnesses; it suffices to provide a Big-O expression, or some expression giving the exact number of scalar multiplications.)

6. {5 marks} How do we go about picking a good hash function and hash structure? Provide some general guidelines about what a hash designer needs to think about.

7. {4 marks} Given the following postfix visitation/traversal of a math expression, draw a valid expression tree for it (similar to the examples from class): $2\ 3\ 4\ *\ -\ 4\ 8\ 2\ /\ +\ +$

8. {2 marks} Explain why quadratic probing is preferable over linear probing, for the open addressing form of hashing.

9. {4 marks} Place the following elements into a nearly complete binary tree. After doing that, build a heap using the Heapify algorithm shown in class. The elements (in level-wise order) are as follows:

10, 6, 5, 17, 4, 22, 3