CSE 3500: Algorithms and Complexity

Homework 1: Due on September 22, 2026, 2PM

1. What is the exact number of executions of the statement `Result++` in the following algorithm? First derive a closed-form expression for the running time, and then simplify it using asymptotic notation.

```
Algorithm Count();
    Result := 0;
    for i := 1 to n do
        for j := 1 to i do
            for k := 1 to j^2 do
                Result++;
    Output Result;
```

2. Show that:

(a)

$$9n^4 \log n + 17n^3 = \Theta(n^4 \log n).$$

(b)

$$\log((2n)!) = \Theta(n \log n).$$

(c)

$$\left(\sqrt{n}\right)^{n^{1/3}} = o\left(2^{n^{0.7}}\right).$$

Give appropriate justification for each answer.

3. Prove or disprove each of the following statements. If a statement is false, give a counterexample.

(a) If

$$f(n) = O(g(n)),$$

then

$$3^{f(n)} = O(3^{g(n)}).$$

(b)

$$\min\{f(n), g(n)\} = \Theta\left(\frac{f(n)g(n)}{f(n) + g(n)}\right),$$

assuming $f(n), g(n) > 0$.

(c) For any real constants a and b,

$$(an + 1)^b = \Theta(n^b),$$

provided $a > 0$.

4. Input is an array $A[1 : n]$ of arbitrary real numbers. It is known that the array contains three special elements, each of which occurs $n/6$ times. Every other element occurs exactly once. Present a Las Vegas randomized algorithm to identify all three repeated elements. The running time of the algorithm should be $\widetilde{O}(\log n)$.

5. Let S be a subset of the integers $1, 2, \ldots, n$. Design a data structure supporting the following operations:

- **INSERT(i)**: Insert i into S. If i is already present, do nothing.
- **DELETE(i)**: Delete i from S, if it is present.
- **FIND-ANY**: Return an arbitrary element of S, or report that S is empty.
- **MEMBER(i)**: Determine whether $i \in S$.

Each operation should take $O(1)$ time, independent of $|S|$.

Describe the data structure and explain why each operation takes constant time.

6. Input is a sequence X of n keys containing many duplicates. The number of distinct keys is $d < n$. Design an algorithm that sorts the distinct keys, and reports each distinct key together with its number of occurrences. The total running time should be $O(n \log d)$. For example, if $X = (8, 3, 8, 12, 3, 5, 8, 5, 12, 3, 20)$, the output should be 3(3), 5(2), 8(3), 12(2), 20(1). Analyze the running time of your algorithm.

7. Solve the following recurrence relations, giving tight asymptotic bounds.

(a)

$$T(n) = \begin{cases} 1, & n \leq 2, \\ 9T(n/3) + n^2 \log n, & n > 2. \end{cases}$$

(b)

$$T(n) = \begin{cases} 1, & n \leq 16, \\ T(\sqrt{n}) + (\log n)^2, & n > 16. \end{cases}$$

8. Given are two sets A and B, containing m and n elements, respectively, drawn from a linear order. Neither set is necessarily sorted, and assume $m \leq n$. Show how to compute $A - B$ and $A \cap B$ in $O(n \log m)$ time. Clearly describe your algorithm and analyze its running time.

9. Let $X_1, X_2, \ldots, X_\ell$ be sorted sequences whose total number of elements is $\sum_{i=1}^{\ell} |X_i| = n$. The problem is to merge these sequences. Assume that you must also identify, for every element in the final merged sequence, which input sequence X_i it came from. Design an algorithm that produces the merged sequence, together with these sequence identifiers, in $O(n \log \ell)$ time. Explain the data structure used and prove the running-time bound.