Computer Science 385
Design and Analysis of Algorithms
Siena University
Spring 2026

Problem Set 3

Group Formation: 4:00 PM, Friday, February 27, 2026
Due: 4:00 PM, Friday, March 13, 2026

You may work alone or in a group of size 2 or 3 on this assignment. However, in order to make sure you learn the material and are well-prepared for the exams, you should work through the problems on your own before discussing them with your partner, should you choose to work with someone. In particular, the "you do these and I'll do these" approach is sure to leave you unprepared for the exams.

This is a smaller problem set, but it's happening concurrently with the first empirical study, and you will likely have questions. It will be difficult if not impossible to complete the assignment if you wait until the last minute. A slow and steady approach will be much more effective.

Submitting

Please submit a hard copy (typeset preferred, handwritten OK but must be legible) for all written questions. Only one submission per group is needed.

Stability of Sorting Algorithms

Question 1: See the definition of a *stable* sorting algorithm on p. 19-20 of Levitin. Describe two circumstances where sorting of data is needed, such that for one of the circumstances it is important that the algorithm used for sorting is stable, and one where it does not matter. (4 points)

Question 2: Explain briefly why bubble sort as we studied in class is a stable sorting algorithm. (2 points)

Question 3: Show by an example that selection sort as we studied in class is **not** a stable sorting algorithm. (2 points)

The remaining questions in this section are related to the $ComparisonCountingSort$, introduced inLevitin Exercise 1.3.1.

Question 4: Show the final values in the array $Count$ as computed during the sorting of the array containing the values

$$8\ 6\ 7\ 5\ 3\ 0\ 9$$

passed in as A. (2 points)

Question 5: Show the final values in the array $Count$ as computed during the sorting of the array containing the values

$$8\ 0\ 0\ 5\ 2\ 8\ 1\ 2\ 3\ 4$$

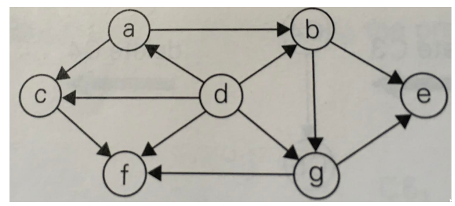
passed in as A. (2 points)

Question 6: This sorting algorithm is not stable. Show this by applying the algorithm to a different example input. Hint: the array doesn't have to be very large. Annotate equivalent input values with their initial positions so you can tell which one is which in the output array S. (3 points)

Topological Sorting

Recall that a topological ordering is a list of a directed graph's vertices in which for every edge in the graph, the vertex where the edge starts comes before the vertex where the edge ends. Below is pseudocode for the Source-Removal Algorithm which computes a topological ordering of the vertices in a directed acyclic graph (DAG).

ALGORITHM SOURCEREMOVAL(G)
  //Input: a dag $G = \{V, E\}$
  $o \leftarrow$ a new empty list
  while $\exists$ a source vertex in G **do**
    $v \leftarrow$ a source vertex from G
    $o.append(v)$
    $G.remove(v)$ // also removes all incident edges
  if G is empty **then**
    return o
  else
    flag error // G was not a dag



Question 7: Apply the algorithm to the DAG above. Give the vertices in the order in which they are added to the list o by the algorithm. Break ties by alphabetical order. (5 points)

Question 8: The Source-Removal algorithm gives one topological ordering of the vertices, but there may be others. For the graph above, give one other valid topological ordering of its vertices. (2 points)

Solving Recurrences

Determine closed forms for each of the following recurrence formulas using backward substitution. (6 points each, total 48 points)

For each problem: (i) show at least the first 3 substitutions (for 2 points), (ii) show the pattern for the i^{th} substitution (1 point), (iii) state the value that i will have in the substitution that allows you to apply the base case ($\frac{1}{2}$ point), (iv) give the closed form (2 points), and (v) give the Θ efficiency class of the recurrence ($\frac{1}{2}$ point). Where appropriate, you must apply an $n = 2^k$ (or similar) substitution.

Question 9: $T(n) = T(n-1) + 5$ for $n > 1, T(1) = 3$

Question 10: $T(n) = T(n-1) + 5n$ for $n > 1, T(1) = 3$

Question 11: $T(n) = T(n/2) + 4$ for $n > 1, T(1) = 1$

Question 12: $T(n) = T(n/2) + 4n$ for $n > 1$, $T(1) = 1$

Question 13: $T(n) = 2T(n/2) + 3$ for $n > 1$, $T(1) = 1$

Question 14: $T(n) = 2T(n/2) + 3n$ for $n > 1$, $T(1) = 1$

Question 15: $T(n) = 2T(n/2) + an$ for $n > 1$ and any positive constant a, $T(1) = 1$

Note: although your final closed form for the problem above will include the constant a, your Θ growth rate should only be in terms of n. Take advantage of the fact that a is a constant.

Question 16: $T(n) = 2T(n/2) + n^2$ for $n > 1$, $T(1) = 1$

Grading

This assignment will be graded out of 70 points.

Feature	Value	Score
Question 1	4	
Question 2	2	
Question 3	2	
Question 4	2	
Question 5	2	
Question 6	3	
Question 7	5	
Question 8	2	
Question 9	6	
Question 10	6	
Question 11	6	
Question 12	6	
Question 13	6	
Question 14	6	
Question 15	6	
Question 16	6	
Total	70	