COSC 2123/1285 Algorithms and Analysis
Tutorial 1
Introduction to algorithms analysis and fundamental data structures

1 Information

- Additional exercises are available in the book if you need more practice.
- Hints are available for each question at the end of the book.

2 Additional Resources

- Check the discussion boards on Canvas to see if your question has been answered. If not, discuss problems on the discussion boards.
- See the lecturer during consultation hours.

---

3 Questions

1.4.3a Show the stack after each operation of the following sequence that starts with the empty stack:

push(a), push(b), pop, push(c), push(d), pop

1.4.3b Show the queue after each operation of the following sequence that starts with the empty queue:

enqueue(a), enqueue(b), dequeue, enqueue(c), enqueue(d), dequeue

1.4.4a Let A be the adjacency matrix of an undirected graph. Explain what property of the matrix indicates that:

a. the graph is complete, i.e., there is an edge between every pair of distinct vertices.

b. the graph has a loop, i.e., an edge connecting a vertex to itself.

c. the graph has an isolated vertex, i.e., a vertex with no edges incident to it.

1.4.4b Answer the same questions as in a. for the adjacency list representation.

1.4.9 For each of the following applications, indicate the most appropriate data structure:

a. answering telephone calls in the order of their known priorities.

b. sending backlog orders to customers in the order they have been received.

c. implementing a calculator for computing simple arithmetical expressions.

+

1.4.2 If you have to search for an element in a list/collection of n numbers, how can you take advantage of the fact that the list is known to be sorted? Give separate answers for:

a. lists represented as arrays.

b. lists represented as linked lists.

1.4.10 Design an algorithm for checking whether two given words are anagrams, i.e., whether one word can be obtained by permuting the letters of the other. For example, the words *tea* and *eat* are anagrams.

