

Discussion 6

Mengxiang Jiang

CSEN 5303 Foundations of Computer Science

September 25, 2022

Problem 1. Discuss the data structure *queue* and its fundamental operations. Give an illustrative example.

A *queue* is an ordered group of same objects such that the first objects added to the group is also the first objects to leave (First In First Out FIFO). The two fundamental operations are *enqueue* and *dequeue*. *Enqueue* adds new objects to the end of the queue while *dequeue* removes elements from the front of the queue.

An illustrative example is suppose a group of students arrive at the Javelina Dining Hall. They enqueue into a line with Alice at the front, then Bob, then Charlie, and finally Danielle at the end. The receptionist scans Alice's id first and dequeues her, allowing her into the dining hall. Then she scans Bob and dequeues him. In the meantime, Eve and Frank arrive and enqueue behind Danielle. The receptionist continues scanning ids and dequeuing from Charlie until Frank and the queue is empty.

Example of the student queue

Step	Operation	Queue
1	Enqueue Alice	Alice
2	Enqueue Bob	Alice, Bob
3	Enqueue Charlie	Alice, Bob, Charlie
4	Enqueue Danielle	Alice, Bob, Charlie, Danielle
5	Dequeue	Bob, Charlie, Danielle
6	Dequeue	Charlie, Danielle
7	Enqueue Eve	Charlie, Danielle, Eve
8	Enqueue Frank	Charlie, Danielle, Eve, Frank
9	Dequeue	Danielle, Eve, Frank
10	Dequeue	Eve, Frank
11	Dequeue	Frank
12	Dequeue	

Problem 3. Indicate whether each of the following applications would be suitable for a queue.

a. An ailing company wants to evaluate employee records so as to lay off some workers on the basis of service time(the most recently hired employees are laid off first).

No, since the most recently hired employees will be at the end of a queue while dequeueing happens at the front, a stack is more suitable.

b. A program is to keep track of patients as they check into a clinic, assigning them to doctors on a first come, first served basis.

Yes, assuming there's no priority for emergency or intensive care, the first come first served nature is perfect for a queue.

c. A program to solve a maze is to backtrack to an earlier position (the last place where a choice was made) when a dead-end position is reached.

No, since the most recent place a choice was made will be at the end of the queue while dequeueing happens at the front (or the oldest decision made), a stack is more suitable.

d. An inventory of parts is to be processed by part number.

Yes, as long as you will enqueue the parts in the order of their part number, then the dequeue for processing will also be in order of part number.

e. An operating system is to process requests for computer resources by allocating the resources in the order in which they are requested.

Yes, the first come first served nature allocating resources fits a queue.

f. A grocery chain wants to run a simulation to see how the average customer wait time would be affected by changing the number of checkout lines in its stores.

Yes, since each checkout line processes the customer's purchase in the order they arrive, they are queues.

g. A dictionary of words used by a spelling checker is to be initialized.

No, a spelling checker needs to have fast ($O(1)$) access to any element in its dictionary, while a queue's access is slow ($O(n)$) for most of its elements.

h. Customers are to take numbers at a bakery and be served in order when their numbers come up.

Yes, since service is in order of first come first serve, it is a queue.

i. Gamblers are to take numbers in the lottery and win if their numbers are picked.

No, the lottery does not award the prize in order of who entered but randomly selects one.