НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Data Structures and Organization

(p.3 – Stacks and Queues)

Yevhen Berkunskyi,

Computer Science dept., NUoS

eugeny.berkunsky@gmail.com

http://www.berkut.mk.ua

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Stacks and Queues

In this lecture we'll see three data storage structures:

- the **Stack**,
- the **Queue**, and
- the **Priority Queue**

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

A Different Kind of Structure

- Programmer's Tools
- Restricted Access
- More Abstract

Programmer's Tools

- Arrays are appropriate for the kind of data you might find in a database application. They're typically used for personnel records, inventories, financial data, and so on—data that corresponds to real-world objects or activities.

- The structures and algorithms we'll see here are more often used as programmer's tools.
  - They're primarily conceptual aids rather than full-fledged data storage devices.
  - Their lifetime is typically shorter than that of the database-type structures.
  - They are created and used to carry out a particular task during the operation of a program; when the task is completed, they're discarded.

Restricted Access

- In an array, any item can be accessed, either immediately – if its index number is known – or by searching through a sequence of cells until it's found.
- In the data structures in this lecture, however, access is restricted: Only one item can be read or removed at a given time (unless you cheat).

- The interface of these structures is designed to enforce this restricted access. Access to other items is (in theory) not allowed.

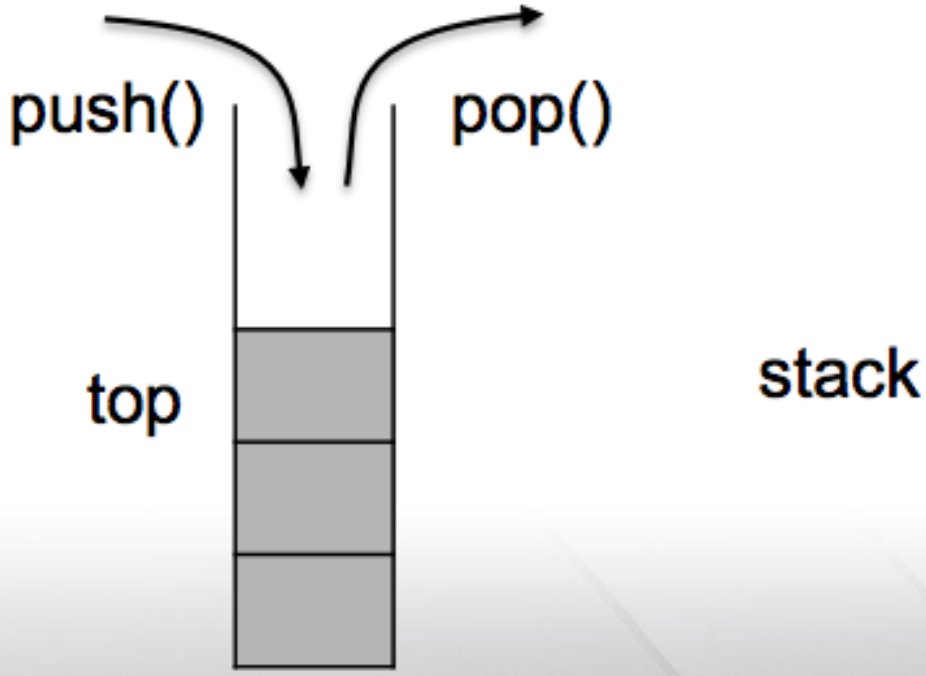
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

More Abstract

- Stacks, queues, and priority queues are more abstract entities than arrays and many other data storage structures.
- They’re defined primarily by their interface: the permissible operations that can be carried out on them.
- The underlying mechanism used to implement them is typically not visible to their user.

Stacks

- A stack allows access to only one data item: the last item inserted. If you remove this item, you can access the next-to-last item inserted, and so on. This capability is useful in many programming situations.



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

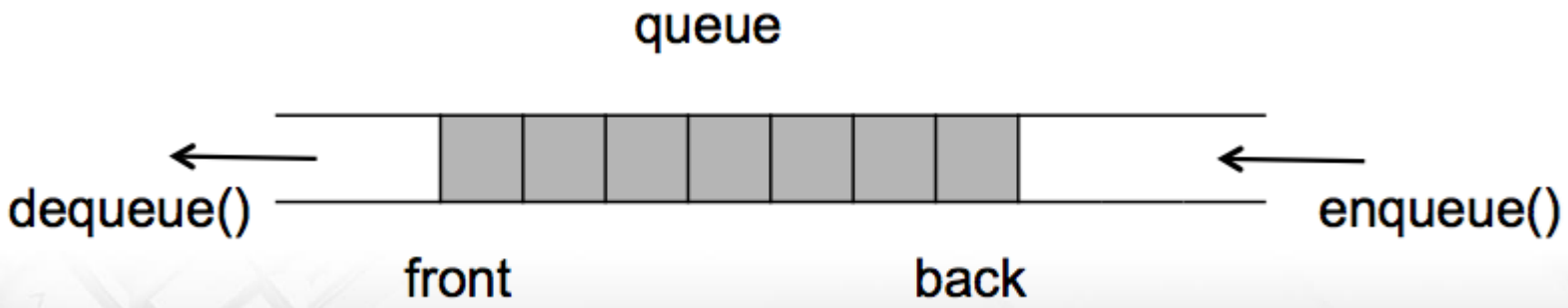
Example

Queues

- In computer science a **Queue** is a data structure that is somewhat like a stack, except that in a queue the first item inserted is the first to be removed (**First-In-First-Out, FIFO**), while in a Stack, as we've seen, the last item inserted is the first to be removed (**LIFO**)

Queues

- Queues are used as a programmer's tool as stacks are.
- They're also used to model real-world situations such as people waiting in line at a bank, airplanes waiting to take off, or data packets waiting to be transmitted over the Internet.

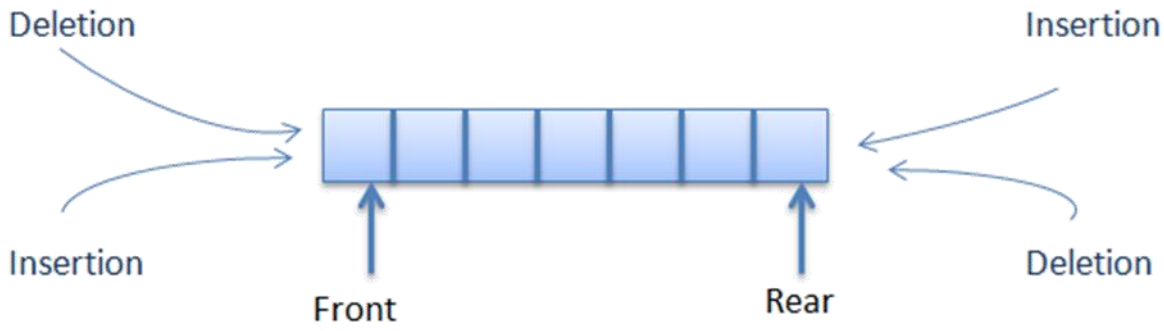


НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Example

Deques

- A ***deque*** is a double-ended queue. You can insert items at either end and delete them from either end.
- The methods might be called **insertFirst()** and **insertLast()**, and **removeFirst()** and **removeLast()** .



Deques

- If you restrict yourself to **insertFirst()** and **removeFirst()** (or their equivalents on the last), the **deque** acts like a **stack**.
- If you restrict yourself to **insertFirst()** and **removeLast()** (or the opposite pair), it acts like a **queue**.
- A **deque** provides a more versatile data structure than either a **stack** or a **queue** and is sometimes used in container class libraries to serve both purposes.

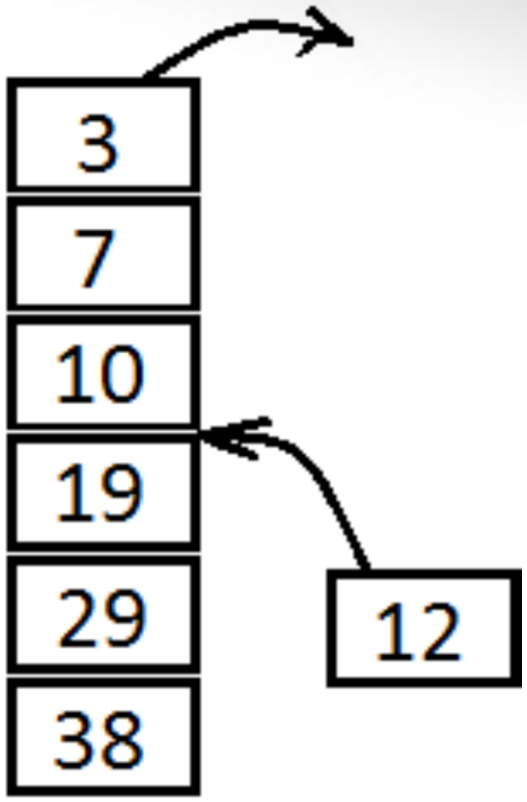
Priority Queues

- A priority queue is a more specialized data structure than a stack or a queue. However, it's a useful tool in a surprising number of situations.
- Like an ordinary queue, a priority queue has a front and a rear, and items are removed from the front.
- However, in a priority queue, items are ordered by key value so that the item with the lowest key (or in some implementations the highest key) is always at the front.
- Items are inserted in the proper position to maintain the order.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Priority Queues

- In many situations you want access to the item with the lowest key value (which might represent the cheapest or shortest way to do something).
- Thus, the item with the smallest key has the highest priority.
- Somewhat arbitrarily, we'll assume that's the case in this discussion, although there are other situations in which the highest key has the highest priority



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Example

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Questions?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Data Structures and Organization

(p.3 – Stacks and Queues)

Yevhen Berkunskyi,

Computer Science dept., NUoS

eugeny.berkunsky@gmail.com

http://www.berkut.mk.ua