

Лекция 1.5. Очереди

Здравствуйте. На этой лекции мы рассмотрим другой тип данных, очередь, и рассмотрим разные способы реализации этого типа.

Определение очереди похоже на определение стека. Очередь – это совокупность однотипных элементов, для которой определены две основные операции:

- добавление элемента в очередь;
- извлечение из очереди элемента, который был добавлен раньше других. В этом отличие очереди от стека.

Тип элементов может быть различным в зависимости от конкретной задачи. Размер очереди (количество элементов) теоретически неограничен, хотя возможны ограничения конкретной реализации.

Таким образом, отличие очереди от стека – только порядок извлечения элементов.

Очередь, как и стек, может быть реализована как массив. Но при этом потребуются две дополнительные переменные: голова очереди, то есть в данном случае позиция в массиве самого «старого» элемента, который должен быть извлечен при очередной операции извлечения, и длина очереди, то есть текущее количество элементов в ней, которая позволяет определить, куда будет помещен следующий добавляемый элемент. Эти два параметра, `head` и `len`, определяют, какая часть массива занята элементами очереди.

Позиции начала и конца очереди будут увеличиваться при операциях добавления и извлечения. Но что произойдет, когда одна из этих позиций достигнет конца массива? Она просто перейдет на начало массива. Такая реализация очереди называется кольцевым буфером. При этом заняты будут края массива, а его середина будет свободна.

Рассмотрим выполнение операций с очередью.

Создание пустой очереди предполагает, что переменным `head` и `len` присваиваются значения 0. Впрочем, это важно только для `len`, параметру `head` можно было присвоить любое другое значение индекса в пределах массива.

При занесении элемента в очередь его позиция `t` определяется как сумма `head + len`, однако сумма берется по модулю `N`, то есть, как было сказано, выполняется переход с конца массива на его начало. Длина очереди `len` при этом увеличивается.

Аналогичным образом при извлечении самого давнего элемента из очереди определяется новая позиция головы. Длина очереди при этом уменьшается.

Проверка очереди на непустоту – это проверка ненулевого значения длины.

Можно реализовать очередь и как список. В данном случае наиболее удобен циклический список с указателем на хвост.

Как ранее было для стека, определение очереди – это определение элемента списка, включающее указатель на следующий (позднее добавленный) элемент и значение текущего элемента. И создание пустой

очереди, как и создание пустого стека, это просто присваивание нулевого значения указателю на очередь.

Непустая очередь выглядит как циклическая очередь, первый элемент которой – это хвост очереди, то есть последний добавленный элемент. За ним в списке следует голова очереди, то есть наиболее давно добавленный элемент.

Посмотрим, как выполняются операции с очередью-списком.

Добавление нового элемента производится в хвост очереди. После того, как создан новый элемент и ему присвоено значение, включение этого элемента в очередь производится несколько разным образом в зависимости от того, пуста ли очередь в этот момент. Чтобы полностью понять особенности выполнения операций, лучше всего нарисовать очередь на бумаге и разобраться с манипуляциями указателями, которые изображаются стрелками..

Извлечение элемента из головы очереди выполняется несколько проще, надо только помнить, что головной элемент – это не тот, на который ссылается указатель q , а следующий за ним элемент t .

Проверка очереди на непустоту – это проверка ненулевого значения указателя q .

Очереди, в отличие от стеков, часто используются именно для описания обрабатываемых данных в различных задачах. К таким задачам относится, например, моделирование поведения различных очередей, возникающих в реальном мире. Это очереди морских судов на погрузку/разгрузку в порту; очереди самолетов, ожидающих свободной посадочной полосы; «пробки», возникающие в дорожном движении.

Другой важный класс задач – это организация взаимодействия модулей сложной программной системы (например, операционной системы): там часто используются очереди сообщений, которые посылаются одними модулями системы для обработки другими модулями.

Итак, что мы узнали о типах данных вообще, о стеках и очередях в частности?

Что очередь – это тип данных, похожий на стек, но отличающийся от стека порядком извлечения элементов.

Что очередь может быть эффективно реализована на основе массивов или списков, причем оба варианта имеют свои преимущества и недостатки.

При этом выбор реализации определяется потребностями конкретной задачи.

Что к числу задач, где используются очереди, относится моделирование очередей в реальном мире, а также реализация очередей сообщений в информационных системах.

На этом лекция закончена. Благодарю за внимание.