Московский авиационный институт
(государственный технический университет)

Факультет «Прикладная математика и физика»
Кафедра «Вычислительная математика и информатика»

Курсовой проект по

Языкам и методам программирования по теме:

«Линейные списки»

Выполнил: Щербаков А.А.
Студент гр. М8О-106Б
Преподаватель: Дубинин А.В.
Оценка:
Дата:

2017 год

Введение

Для хранения некоторой информации человеку свойственно ее упорядочивать. Работа с информацией, которая находится в беспорядке, неудобна и слишком затратна. Так и работа компьютера с некоторой информацией будет более быстрой и удобной при ее некотором едином представлении. Для множества целей были созданы некоторые структуры данных, работа с которыми в определенных случаях приносит ощутимую пользу. Такими структурами являются, например, массивы, векторы, стеки, очереди, деки, списки. Каждая структура данных имеет достоинства и недостатки, которые определяют сферу, в которых она может быть использована. Списки имеют множество вариаций, которые также могут быть применены в различных условиях.

Списки

Наиболее используемым вариантом списка является связный список - базовая динамическая структура данных, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну или две ссылки на следующий и предыдущий узел списка.

Преимуществом списка перед, например, массивом является представление данных в памяти компьютера – данные не обязательно идут друг за другом. Также добавление и удаление элемента происходит за O(1), а не O(N), в случае массива.

Основные виды списков:

1. Линейный – каждый элемент указывает на предыдущий/следующий, имеет начало и конец:
    a. Однонаправленный (с барьерным элементом и без)
    b. Двунаправленный (с барьерным элементом и без)
2. Кольцевой – каждый элемент указывает на предыдущий/следующий, а последний элемент на первый (первый на последний):
    a. Однонаправленный
    b. Двунаправленный
3. XOR-список – имеет только один адрес – результат выполнения операции XOR над адресами предыдущего и следующего элементов списка.

Структура этих списков схожа – каждый список состоит из указателя на начало, указателя на следующий/предыдущий и размера списка. Элемент списка имеет указатель на следующий/предыдущий и значение данного элемента. Значение указателя последнего элемента линейного списка равно NULL. Такое представление данных позволяет:

1. Добавлять и удалять элементы списка за O(1)
2. Хранить данные списка в памяти неупорядоченно

В то же время имеются некоторые недостатки:

1. Доступ к элементу за O(N), где N – размер списка
2. Необходимость использования итератора – элемента, с помощью которого происходит перемещение по элементам списка.

Итератор представляет собой указатель на текущий элемент списка. К нему определены некоторые действия: перемещение по элементам списка и извлечение/изменение значения текущего элемента.

Задача — переставить местами две половины линейного однонаправленного списка.

Алгоритм:

1. Создать два итератора в начале списка((1) и (2)).

2. Сдвинуть итератор (2) на величину (длина списка)/2. Если длина списка нечетная, то сдвинуть еще на один элемент.

3. Пока итератор (2) не находится в конце списка, поменять значения элементов, на которые указывают (1) и (2) итераторы. Сдвинуть оба итератора вправо.

Сложность алгоритма O(N), где N — длина списка.

Заключение

Помимо простого хранения некоторой упорядоченной информации в списке представляется возможным использовать их как основу для других динамических структур данных – стека, очереди, дека, деревьев, графов и т.д. В отличие от реализации подобных структур на массиве, их реализация на списке помогает решить проблему использования памяти в компьютере. Так элементы массива обязаны находиться в памяти друг за другом, что затрудняет их использование в системе, где свободные ячейки памяти разбросаны в ней неравномерно. Список дает возможность избежать этой проблемы – память под элементы списка выделяется независимо от расположения предыдущих элементов (зависимо только в том смысле, что не может быть выделена в уже занятой ячейке памяти), в свободном месте. Таким образом, список является удобной структурой данных при работе с кластеризованной информацией с частым добавлением и удалением элементов (теория графов).

Список источников

1. ru.wikipedia.org/wiki/Список_(информатика) «Теоретическая информация о списках»