

Линейные контейнеры

Д. В. Луцив

Кафедра системного программирования СПбГУ



CS103

Содержание

- 1 Указатели
- 2 Массивы
 - Одномерные
 - Многомерные, мягкие и массивы массивов
- 3 Динамические линейные контейнеры
 - Свойства
 - Списки
 - Прочие

Указатель

- м.б. многокомпонентный (с сегментом, например), что затрудняет сравнение
- м.б. в разной степени относительный
- может иметь фактически неиспользуемые младшие биты (битовая упаковка)

Ссылочные и контейнерные типы

- контейнерный тип
 - при присваивании копируется значение
 - выделяется обычно на стеке или в «сегменте статических данных»
 - легко оптимизируется компилятором
 - пример — `C char`
- ссылочный тип
 - при присваивании копируется ссылка
 - выделяется обычно в куче
 - тяжело оптимизируется кем угодно
 - пример — строка `C`

Минимум

Необходимый минимум:

- адрес 0 (или 1) элемента
- размер одного элемента

Чуть больше:

- 1 или 2 границы массива

Минимум

Необходимый минимум:

- адрес 0 (или 1) элемента
- размер одного элемента

Чуть больше:

- 1 или 2 границы массива

Языки без структур данных

- Fortran 77
- Классический BASIC

Массив структур → несколько массивов.

Языки без структур данных

- Fortran 77
- Классический BASIC

Массив структур → несколько массивов.

Паспорт многомерного массива

Служебная структура данных, содержащая для каждого измерения:

- адрес элемента с индексом 0
- возможно только во время компиляции
 - границы измерения
 - размер элемента

Паспорт многомерного массива

Служебная структура данных, содержащая для каждого измерения:

- адрес элемента с индексом 0
- возможно только во время компиляции
 - границы измерения
 - размер элемента

Хранение многомерного массива

- Левый индекс старший (семейство C)
- Правый индекс старший (Фортран)

Вектор Айлифа (массив ссылок на массивы)

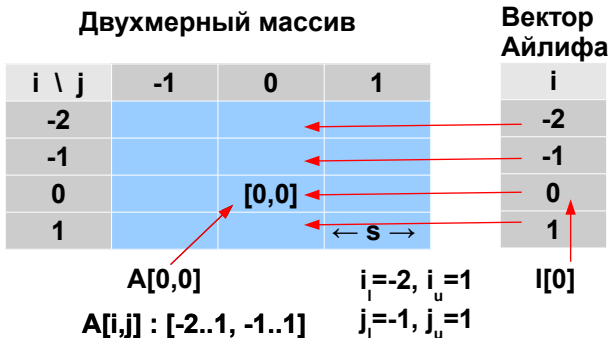
Дж. Айлифф, 1961. [▶ В Википедии](#).

Из современных ЯВУ используются в C (`int**` или `int*[]`, но не `int[][N]`), Java, Perl, Python, Ruby.

В Java массив — всегда ссылочный тип, в Pascal — контейнерный. В C и C# — по-разному, в Perl, Python, Ruby — массивов, как таковых, нет, вместо них списки.

- используются для реализации многомерных массивов
- массив хранит ссылки на массивы следующего уровня
- позволяют отказаться от умножения
- позволяют делать массивы нижних уровней переменной длины

Многомерный массив и вектор Айлифа



$$A[i,j]^{\wedge} = A^{\wedge} + s * ((i_u - i_l + 1) * i + j);$$

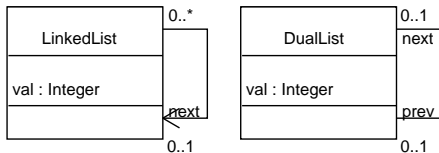
$$I[i]^{\wedge} = I + sp * i; \quad A[i,j]^{\wedge} = I[i] + s * j;$$

Массив

Является встроенной структурой данных большинства типичных алгоритмических императивных языков. Будет рассмотрен, в связи с этим, в соответствующем разделе.

- Однороден по размеру и, обычно, природе элементов;
- хорошо поддерживается аппаратурой и оптимизируется компиляторами;
- не позволяет, в общем случае, изменять свой размер. Сначала он слишком пустой, потом слишком полный;
- не имеет топологии – данные идут подряд, при реструктуризации необходимо их все перемещать.

Список



В зависимости от реализации может быть в различной степени избыточен по памяти.

Курсор/итератор

Атомарный объект или структура для:

- доступа к текущему элементу контейнера
- сдвига вперёд и/или назад
- иногда — доступа к элементу по индексу

Курсор массива — часто прост указатель.

Курсор/итератор

Атомарный объект или структура для:

- доступа к текущему элементу контейнера
- сдвига вперёд и/или назад
- иногда — доступа к элементу по индексу

Курсор массива — часто прост указатель.

Топологии

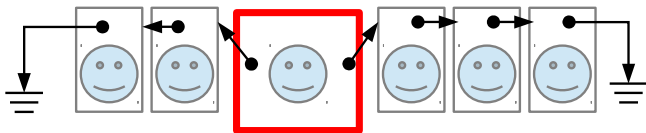
- Односвязный. Навигация только от головы к хвосту, избыточная структура;
- шитый. Навигация от головы к хвосту, но с указателями на далеко стоящие элементы;
- двусвязный. Требует изменяемых структур данных, не поддерживается некоторыми языками (Caml, Haskell). Навигация в обе стороны.
- Zipper. Компромиссный вариант. Навигация в обе стороны, но односвязный. Как молния с двумя замками.

Реализация

При помощи:

- Указателей. В структуре выделяется место под адрес следующего (и, для двусвязного, предыдущего) элементов.
- Массива. Элементы лежат подряд. Для вставки и удаления – сдвиг. Совсем избыточная структура. Быстрая индексация.
- Массива с курсорами. Курсоры – индексы массива, имитирующие указатели. Требуется самостоятельно реализовывать алгоритмы выделения и освобождения памяти. Для языков без структур данных — отдельные массивы для данных списка и для топологии.
- Ленивых генераторов (односвязный) – `comprehension`.
- Обеспечения доступа к элементам через итераторы (могут учитывать оптимизации при перестройке списков).

Zipper

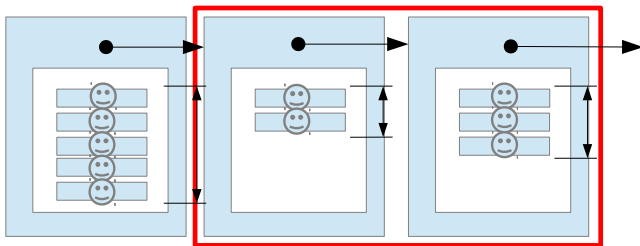


Оптимизация

При помощи:

- Хранения по нескольким элементов в объекте (блочный список). Это уменьшает количество объектов и разгружает кучу, но при редактировании списка вызывает фрагментацию и может потребовать дефрагментации (с потерей актуальности указателей, которые при такой модели вообще не очень уместны, а уместны сложные итераторы).
- Крайние элементы списка можно хранить в локальных переменных. Это ускорит доступ к началу (и, для двусвязного, концу) списка и позволит не выделять динамически память для коротких списков.
- При использовании неизменяемых структур данных для получения списка, начинающегося с середины имеющегося, не надо копировать данные.

Блочный список



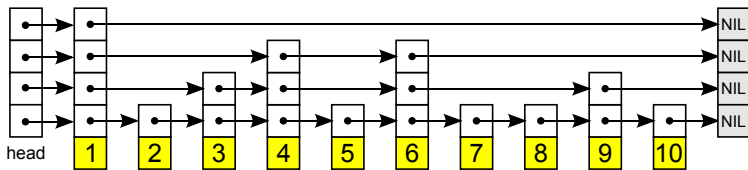
Продвинутая оптимизация: с пропусками I

Отсортированные списки продвинуто оптимизируется при помощи иерархических пропусков. [▶ См. 1](#) [▶ См. 2](#)

Строится иерархия шитых списков (сразу много ссылок на следующий): каждый следующий список P_i будет содержать каждый k -й элемент предыдущего списка P_{i-1} .

- Поиск происходит за $\log_k N$.
- Вставка:
 - 1 ищем, куда вставить, вставляем в нижний список;
 - 2 с вероятностью $\frac{1}{k}$ повторяем для предыдущего списка, с вероятностью $1 - \frac{1}{k}$ заканчиваем работу;
 - 3 если не дошли до верха, то повторяем п. 2.
- Удаление: просто удаляем элемент отовсюду.

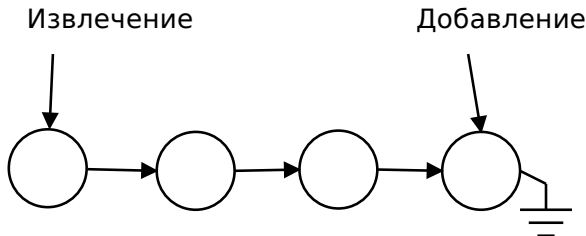
Продвинутая оптимизация: с пропусками II



Очередь

Реализуется при помощи:

- Двусвязного списка;
- односвязного списка с указателем на последний элемент (неизбыточна);
- циклического массива. Одного или нескольких (в аппаратуре – DMA).



Стек

Фактически, односвязный список, со всеми его частностями. И больше ничего. Два стека – Zipper.

В языках программирования без поддержки рекурсии (даже первые версии Си) применялся для имитации рекурсии вручную. На многих архитектурах «с историей», типа IBM3x0, до сих пор аппаратно не поддерживается.

Дек

Структура данных, подразумевающая просмотр, добавление и удаление элементов с обоих концов. Реализуется при помощи, как минимум, двусвязного списка.

Вопросы



▶ EDU.DLUCIV.NAME