

Отчёт

Э.В. Кораблина, Э.Е. Обоскалова

14 июля 2017 г.

Оглавление

1	Разрежённый массив	3
1.1	Введение	3
1.2	Проблема	3
1.3	Решение	4
1.4	Результаты	4

Аннотация

Разрежённый массив — массив, в котором полезные данные (отличные от уникального значения) занимают малую часть от его реального размера. Полезные данные хранятся в структуре индекс-значение, для всех остальных элементов возвращается значение по умолчанию. Простейшая реализация этого массива выделяет место под весь массив, однако когда значений отличных от значений по умолчанию мало, такая реализация неэффективна. К этому массиву не применяются функции для работы с обычными массивами в тех случаях, когда о разрежённости известно заранее, например, при блочном хранении данных.

Глава 1

Разрежённый массив

1.1 Введение

Хранить разрежённый массив в качестве списка куда эффективнее, чем использовать традиционный массив. Для начала дадим определение списка. Связный список — базовая динамическая структура данных в информатике, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну или две ссылки («связки») на следующий и/или предыдущий узел списка. Принципиальным преимуществом перед массивом является структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями. Линейный двунаправленный список — структура данных, хранящая элементы одного типа, связанных между собой указателями. Каждый узел такого списка имеет две связи : с последующим и предыдущими элементами. Ссылка «назад» первого элемента как и ссылка «вперед» последнего элемента указывает на NULL. Первый элемент, на который нет указателя называется «головой» списка, последний «хвостом». В двусвязном списке можно передвигаться вперед и назад. Такой вид списка позволяет узнать адрес предыдущего и последующего элементов, опираясь на содержимое текущего узла. Это позволяет реализовать последовательный доступ к элементам. Реализация разрежённого массива как двусвязного списка используется потому что, во-первых, обычный массив требует место для хранения неопределённых значений. Это повлечет за собой пустую трату памяти. В случае списка значения по умолчанию не хранятся, и место под новые значения выделяется автоматически при добавлении и удалении элементов, то есть можно говорить о динамическом выделении памяти. В главе 1.2 описывается постановка задачи. В главе 1.3 описаны методы решения поставленной задачи. В главе 1.4 дается оценка проведенной работе.

1.2 Проблема

Итак, нам дан массив вещественных чисел большого размера (INT_MAX). Надо заполнить его случайным количеством элементов, результат вывести в файл.

1.3 Решение

INT MAX— максимальное значение для `int` равно 2 147 483 647. Большое количество чисел займет много места в памяти устройства. Так как значения, записываемые в файл, случайны, то велика вероятность, что какое-то из чисел будет повторяться. Следовательно, есть смысл использовать разреженный массив, выбрав наиболее повторяющийся элемент. Предположим, что это ноль. Разреженный массив реализуем на базе двусвязного списка. Поля узла списка: ссылки на предыдущий и последующий элементы, а так же значение позиции ненулевого элемента и его ключ-значение. Для класса разреженный массив на понадобится стандартный набор из конструктора по умолчанию, конструктора копии и деструктора и основных операций: получение длины массива, количества не пустых элементов, вывод не пустых элементов массива, метод `foreach`, применяющий функцию обработки для всех не пустых элементов массива. Перегрузим операторы ввода/вывода и индексации. Поля класса—размер массива и значение элемента по умолчанию.

Итак, нам дан пустой массив вещественных с заданным размером. Заполним его случайными значениями, используя функцию `rand`. Уникальное значение для данного массива будет равно нулю. Из этого следует, что все элементы равные нулю, не попадут в файл с именем `array.txt`. Функция `toFile` перенесет заполненный массив в файл.

В Приложение 1 представлены основные функции класса(конструкторы и деструктор). В Приложение 2 методы и основные операции(печать, ввод и вывод в поток, индексация, метод `for_each`, получение реального размера массива и количества не пустых элементов). В Приложение 3 функции для решения поставленной задачи(`toFile`). В Приложение 4 вспомогательные функции(`add_five`, `copy_list`, `delete_list`, `compare`, `compare_l`, `test_l`, `solve_task`). В Приложение 5 тесты, проверяющие корректность имеющихся функций (`t_length`, `t_count_elem`, `t_for_each`, `index`, `t_compare`, `t_copy`).

1.4 Результаты

В результате был выполнен задачи был получен файл `array.txt`, который содержит ННН элементов. ННН вводимых элементов оказались нулевыми, поэтому в конечный файл не попали. Общее время выполнения ННН. Данный проект максимально покрыт тестами. То есть можно говорить о корректной работе всех функций.

Литература

[1] https://ru.wikipedia.org/wiki/Разрежённый_массив

[2]

[3]

Приложение 1

Приложение 2

Приложение 3

Приложение 4

Приложение 5