

Программирование

Сергей Салищев

Тема 4. Структуры данных на основе массива

Обзор

- Многомерный массив
- Динамический массив
- Циклический буфер
- Куча
- Отсортированный массив
- Хеш таблица с открытой адресацией

Многомерный массив

- Используется:
 - Плотная матрица, изображение
- $f(i_0, i_1, \dots, i_N) \rightarrow i, i_j \text{ in } 0..(K_j-1)$
- $f = i_0 + i_1 * K_0 + \dots + i_N * K_0 * \dots * K_{N-1}$

Динамический массив

- Размер заранее не известен
- При записи за пределы текущего размера — выделение большего блока памяти и копирование
- При мультипликативном росте размера среднее время доступа — $O(1)$
- Реализация: `ArrayList`, `Vector`

Циклический буфер

- Используется: очередь, магазин
- Индексирование по модулю длины буфера
- Часто совмещается с динамическим массивом
- Реализация: `ArrayList`, `ArrayBlockingQueue`

Пирамида

- Используется: очередь с приоритетами
- Также используется для сортировки

Отсортированный массив

- Бинарный поиск
- Скорость сортировки сравнением
- Сортировка слиянием
- Быстрая сортировка
- Сортировки за N^2
- Комбинированные сортировки
- Сортировка за линейное время
 - Цифровая сортировка
 - Сортировка вычерпыванием
 - Сортировка подсчетом

Бинарный поиск

- Алгоритм типа «разделяй и властвуй»
- Эквивалент поиска в сбалансированном бинарном дереве

Скорость сортировки сравнением

- Сортировка — выбор подстановки + сортировка подстановки
- Сортировка подстановки — $O(N)$
- Выбор подстановки — путь от корня до листа в бинарном дереве с подстановками в вершинах
- Высота сбалансированного дерева — $\log N!$
- Формула Стирлинга - $N! \sim \sqrt{2\pi N} (N/e)^n$

Сортировки за N^2

- Сортировка вставками
- Сортировка выбором
- Сортировка пузырьком

Сортировка за $N \log N$

- Сортировка слиянием
- Быстрая сортировка
- Пирамидальная сортировка

Комбинированные сортировки

- Устойчивость сортировки
- Все библиотеки используют сортировку вставками для малых сегментов
- Introsort — гибрид Heapsort и Quicksort анализирующий глубину рекурсии (STL C++)
- Timsort — Mergesort с анализом сегментов (Python)

Цифровая сортировка

- Представим элементы как N разрядные числа
- Отсортируем по разрядам начиная с младшего
- Частичная сортировка должна быть устойчивой (*stable*)
- Сортировка слиянием устойчива

Сортировка вычерпыванием

- Разобьем интервал допустимых значений на сегменты
- Распределим элементы по сегментам
- Отсортируем каждый сегмент
- Объединим сегменты

Таблица с прямой адресацией

- Значение элемента как индекс
- Пример: сортировка подсчетом

Хеш таблица с открытой адресацией

- Таблица с прямой адресацией
- Хеш функция
- Разрешение коллизий при открытой адресации
- Анализ времени работы

Хеш функция

- Ключи как числа
- Композиция хеш-функций
- Метод деления
- Метод умножения

Алгоритм	Псевдокод
Boxcar	$hash += b$
MLCG	$hash = hash * A + b$
rolN-xor	$hash = ROL(hash, N) \wedge b$
rolN-xor[]	$hash = ROL(hash, N) \wedge A[b]$
Xor	$hash \wedge= A[b]$
xAdler	$s1 += b; s2 += s1; hash = s1 \wedge s2; hash \wedge= hash \gg 6;$
Rabin	$hash = ((hash \ll 8) b) \wedge A[hash \gg N]$

Разрешение коллизий при открытой адресации

- Линейное исследование
- Двойное хеширование

Вопросы?