

Цель работы

1. Реализовать структуры данных:
 - «левацкая» куча,
 - «косая» куча.
2. Реализовать общие методы ускорения сливаемых куч:
 - вставка за $O(1)$,
 - bootstrapping.
3. Проверить корректность и эффективность реализованных структур.

Реализация

Интерфейсы

Определен класс `Data.Heap.MergeHeap` сливаемых куч, поддерживающий операции:

- `empty`
- `singleton`
- `insert`
- `getMin`
- `extractMin`
- `merge`
- `build`

Структуры данных

Две структуры данных — `Data.Heap.LeftistHeap` и `Data.Heap.SkewHeap` — реализуют указанный выше класс.

Конструктор типов `Data.Heap.FastIns` позволяет ускорить операцию вставки любой кучи, реализующей класс `MergeHeap`; полученная в результате куча также реализует этот класс.

Конструктор типов `Data.Heap.Bootstrap` позволяет ускорить операцию слияния любой кучи, реализующей класс `MergeHeap`; полученная в результате куча также реализует этот класс.

Тестирование корректности

Для тестирования корректности реализации кучи с её помощью осуществляется сортировка списка посредством поочередной вставки и извлечения всех его элементов,

Программа-тестировщик находится в файле `Tester.hs`, там же можно выбрать кучу для тестирования, задать размер сортируемого списка и число испытаний. Запуск тестирования осуществляется командой `make test`.

Тестирование производительности

Для измерения производительности потребуется пакет `Criterion`, а для построения графиков — консольная утилита `gnuplot`.

Первый тест измеряет скорость сортировки случайных списков различной длины. Для построения графика используется команда `make sort.png`.

Второй тест измеряет скорость слияния двух куч различного размера. Для построения графика используется команда `make merge.png`.

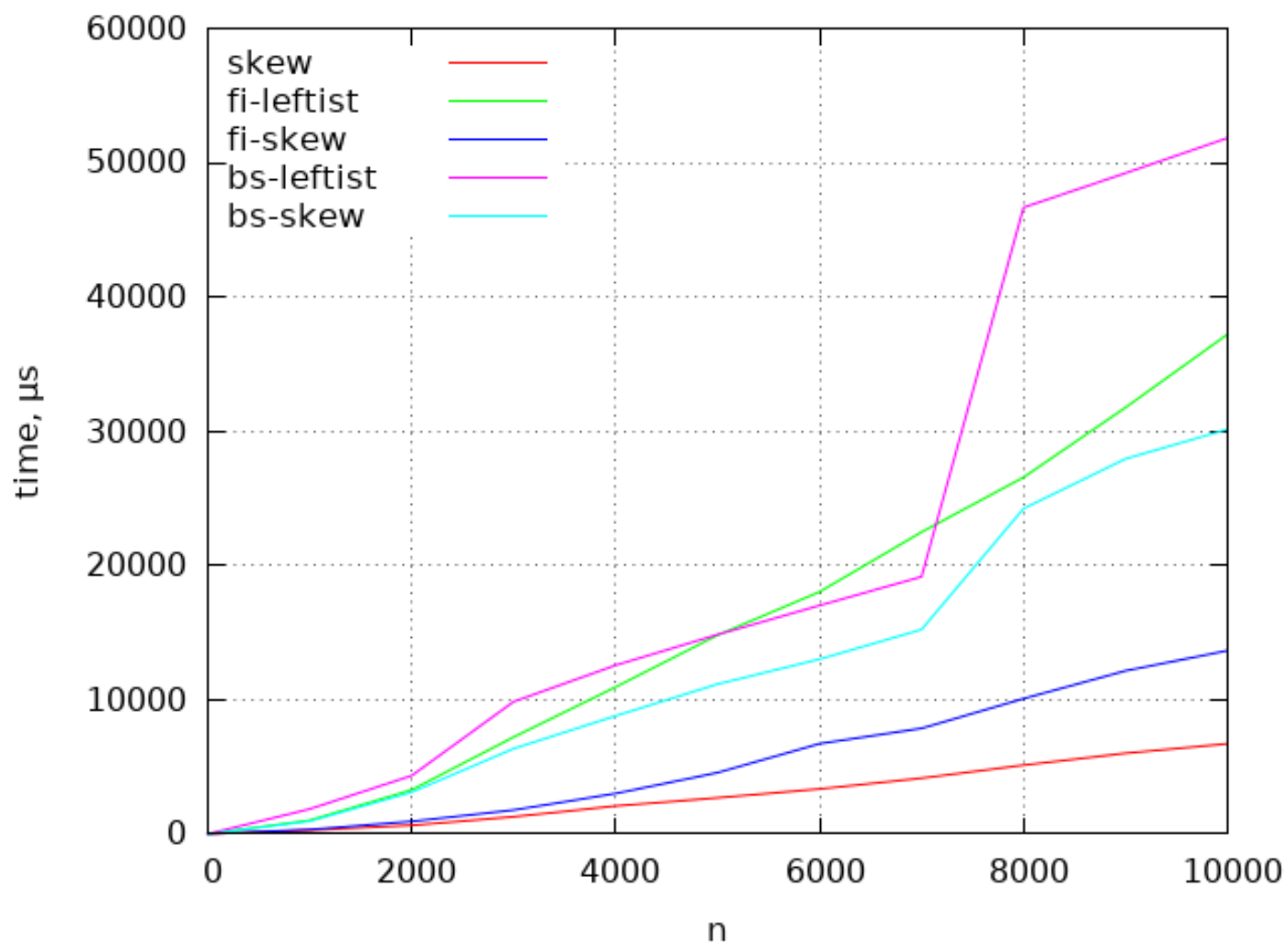


Рис. 1: Сортировка списка

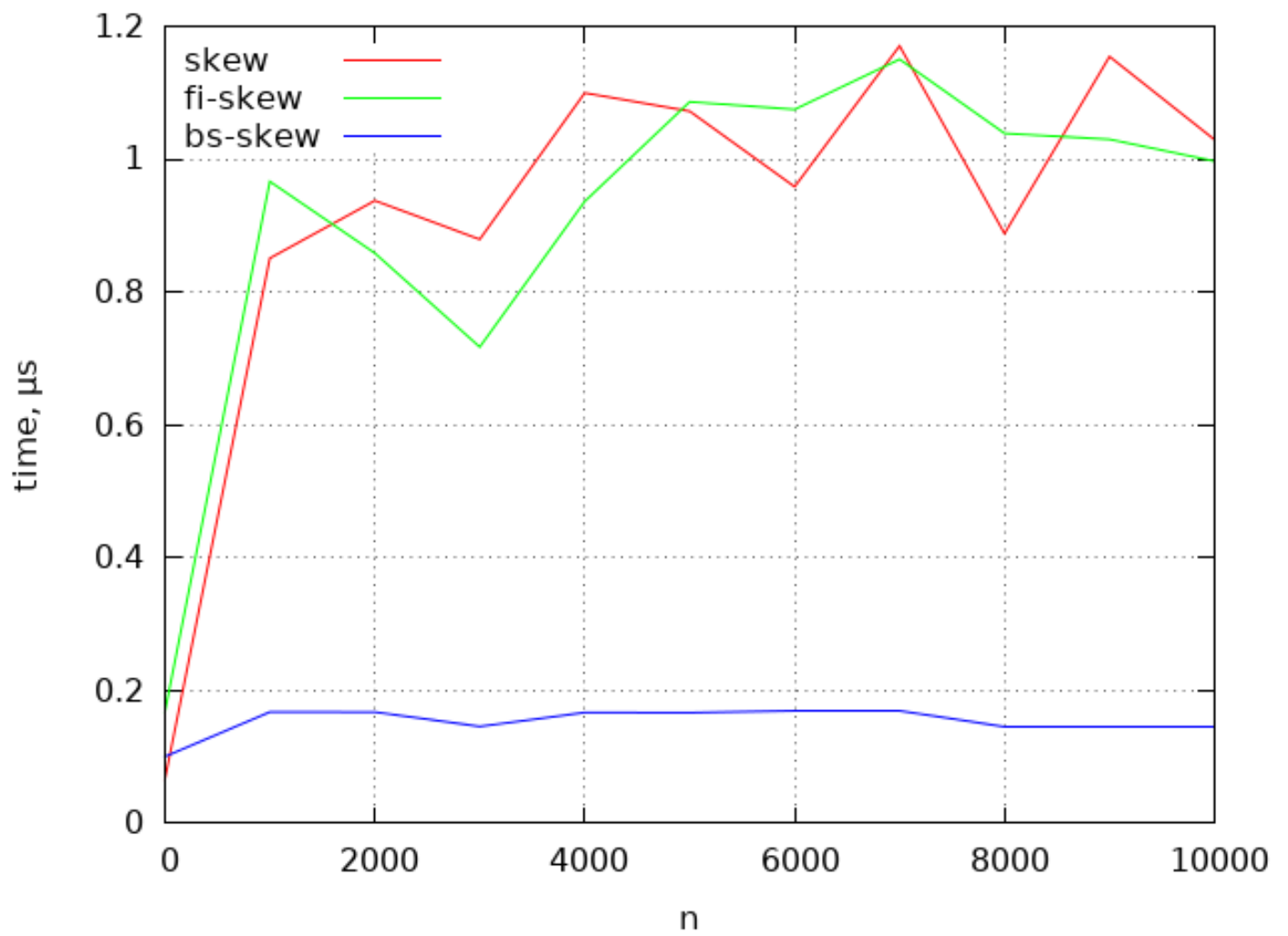


Рис. 2: Слияние двух куч