

# Реализация слабой кучи (weak heap).

Андрей Козлов

Январь 2015

## Постановка задачи

*Слабая куча* — структура данных, позволяющая выполнять операции обычной двоичной кучи с той же асимптотикой, но совершающая при этом меньше сравнений.

Данное свойство может быть полезно, если операции сравнения объектов являются “дорогими”, и требуется минимизировать их количество.

Работа состоит из следующих этапов:

1. реализованная двоичной и слабой куч;
2. проверка корректности;
3. сравнить число операций сравнения при использовании разных куч.

## Реализация

Реализованы два параметризуемых класса `BinaryHeap` и `WeakHeap`, обладающие следующим интерфейсом:

```
public interface IHeap<T extends Comparable<T>> {  
  
    int size();  
  
    boolean isEmpty();  
  
    T findMin();  
  
    T extractMin();  
  
    void insert(T value);  
  
    void build();  
}
```

## Тестирование

Проверка корректности состоит из тестов на следующие операции:

1. поочередное добавление случайных чисел из массива с поиском минимума на каждом шагу;
2. построение кучи из данного массива случайных чисел с поиском минимума на каждом шагу;
3. поочередное извлечение минимума из построенной кучи.

Тесты проводились одновременно на обеих структурах, что позволило на каждой итерации сравнивать полученные значения минимумов друг с другом и с результатом наивного алгоритма, использующего методы класса `java.util.Collections`.

## Сравнение производительности

В качестве “дорогой” операции сравнения использовалась операция сравнения строк из ста символов латинского алфавита.

heap type / <code>compareTo</code> method calls count	1e3	1e4	1e5	1e6
binary	18841	255290	3219373	38794449
weak	9509	128462	1617733	19480089

Заметим, что количество сравнений при использовании слабой кучи сокращается примерно в два раза.