

CS-Club, осенний семестр 2014, курс алгоритмов

Реализация бинарной кучи

Сергей Копелиович

Собрано 14 ноября 2014 г. в 18:12

1. Постановка задачи

Реализовать бинарную кучу.

Протестировать корректность реализации.

Протестировать скорость работы.

Сравнить с `set<int>`, `priority_queue<int>`

2. Реализация

Реализован следующий основной интерфейс:

```
void build( int an, POINTER a ); //  $O(n)$ 
void add( int x ); //  $O(\log n)$ 
T extractMin(); //  $O(\log n)$ 
T getMin();  $O(1)$ 
```

И две внутренние дополнительные функции:

```
void siftUp( int i ); //  $O(\log i)$ 
int siftDown( int i ); //  $O(\log(n/i))$ 
```

3. Тестирование

Тест #1: генерируем случайный массив из n чисел, строим по нему кучу, достаем по очереди элементы из кучи. Замеряем время работы. Сравниваем результат работы решения с результатом наивного решения, вынимающего минимумы из массива за $\mathcal{O}(n)$.

Тест #2: генерируем случайный массив из n чисел, добавляем по одному элементы в кучу, достаем по очереди элементы из кучи. Сравниваем результат работы решения с результатом наивного решения, вынимающего минимумы из массива за $\mathcal{O}(n)$.

Чтобы воспроизвести тесты, запустите `bash run.sh`

4. Результаты тестирования

H – наша реализация бинарной кучи

Q – `priority_queue<int>`

S – `multiset<int>`

n/algo	H	Q	S
10^6	0.61 sec	0.50 sec	1.64 sec
10^7	11.14 sec	10.13 sec	26.82 sec