

# CS-Club, осенний семестр 2014, курс алгоритмов. Orthogonal Range Query 2D implementation with fractional cascading

Ivanov A.K.

20 декабря 2014 г.

## 1 Постановка задачи

Реализовать структуру данных, содержащую мн-во точек на плоскости и поддерживающую запрос:

"выдай все точки из прямоугольника:  $(x_0, y_0, w, h)$ "

протестировать корректность.

протестировать скорость в сравнении с наивным алгоритмом.

## 2 Реализация

Интерфейс:

```
public interface ORQ2D {  
    P[] query(int x0, int y0, int w, int h);  
}
```

Основные методы:

```
// 'points' should be a set  
public FractionalCascadingORQ2D(P[] points)  
// recursive tree generator  
XNode generate(P[] points_x, int l, int r, P[] points_y)  
  
//  $O(\log(n))$  — total time on each query  
List<P> addLeft(XNode n, int ptr)  
List<P> addRight(XNode n, int ptr)  
  
//  $O(\log(n) + k)$  — total time on each query  
List<P> collectFromTheLeft(XNode n)  
List<P> collectFromTheRight(XNode n)
```

## 3 Тестирование

1. Генерируем  $N=2^i * 10$  равномерно распределенных по плоскости точек со значениями координат в  $[0 \dots 10^6]$ .
2. Генерируем 200 случайных запросов вида  $(x_0, y_0, w + 1, h + 1)$  где каждое значение из диапазона  $[0 \dots 10^6]$ .
3. Усредняем время работы каждого алгоритма.

## 4 Результаты тестирования

n/ algo	FC	Naive
81920	0.45 ms	0.59 ms
655360	4.22 ms	12.38 ms
1310720	12.83 ms	20.39 ms
5242880	57.20 ms	96.78 ms

Версия с  $w, h \in [1 \dots 30]$

n/ algo	FC	Naive
327680	0.03 ms	1.58 ms
655360	0.03 ms	4.5 ms
1310720	0.03 ms	10.38 ms

Версия с  $w \in [1 \dots 1000000]$  и  $h \in [1 \dots 10]$

n/ algo	FC	Naive	Binary
163840	0.01 ms	1.1 ms	0.44 ms
327680	0.02 ms	2.9 ms	0.96 ms
655360	0.02 ms	6.6 ms	2.54 ms