

Range Minimal Query

Булгаков Илья, Гусев Илья

Московский физико-технический институт

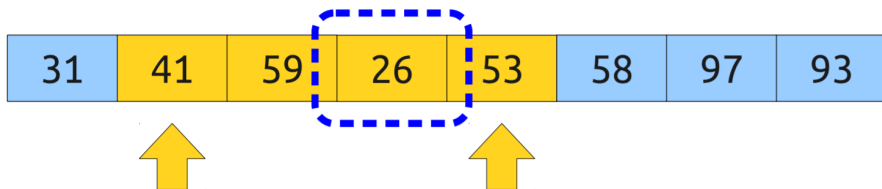
Москва, 2023

- 1 Задача RMQ
 - Постановка и разновидности задачи RMQ
- 2 RMQ: static online
 - Тривиальное решение задачи RMQ (static online)
 - Sqrt-декомпозиция
 - Sparse table

Задача RMQ

RMQ - Range Minimum (Maximum) Query - задача поиска минимума на отрезке.

Дан массив чисел, к нему делаются запросы на поиск минимума на отрезке $[l, r]$



Задача RMQ

Разновидности задач

По количеству запросов:

- offline - можно получить много запросов, проанализировать их все и выдать ответ на все сразу
- online - обработка запросов строго по одному

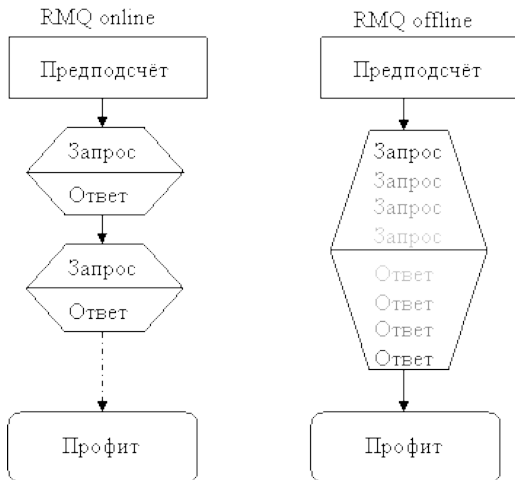
По возможности изменения исходного массива:

- static - массив чисел закреплён
- dynamic - массив чисел меняется

Мы сегодня будем рассматривать **static online** и немного поговорим про подходы к обработке модификаций(**dynamic online**).

Задача RMQ

Online vs offline



Тривиальное решение задачи RMQ

1 Без предобработки

Время препроцессинга: $O(1)$

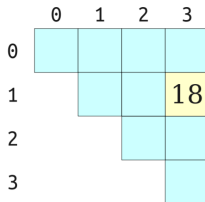
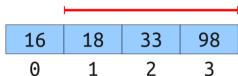
Время ответа: $O(n)$

2 С предобработкой

Время препроцессинга: $O(n^3)$

Время ответа: $O(1)$

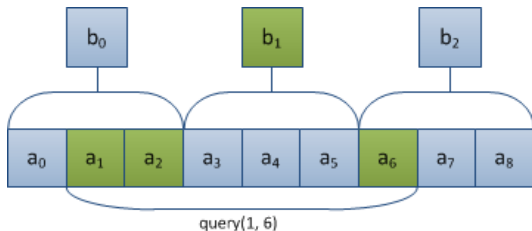
Нужно вычислить n^2 возможных отрезков, каждый за n .



Замечание: n^3 может быть сокращен до n^2 с помощью динамики

Sqrt-декомпозиция

Поделим массив на блоки размером \sqrt{n} . Предподсчитаем минимумы на этих блоках.



При запросе берём минимум из минимумов полностью покрытых блоков и оставшихся элементов неполностью покрытых блоков.

Sqrt-декомпозиция

Время препроцессинга: $O(n)$

Время ответа: $O(\sqrt{n})$

Sparse table

Мотивация

Проблема: время на ответ у sqrt-декомпозиции все еще очень долгое.
Хотим: немного увеличим время на препроцессинг, но добьемся константного времени на ответ.

Цель:

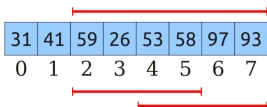
Препроцессинг - $O(n \log(n))$,

Запрос - $O(1)$

Sparse table

Интуиция

Нам не обязательно вычислять все n^2 интервалов, чтобы уметь отвечать про минимум за $O(1)$. Каждый интервал может быть накрыт двумя интервалами длины степени двойки.



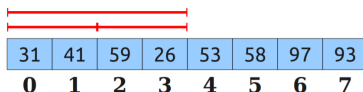
	0	1	2	3	4	5	6	7
0	31	31	31	26				
1		41	41	26	26			
2			59	26	26	26		★
3				26	26	26	26	
4					53	53	53	53
5						58	58	58
6							97	93
7								93

Sparse table

Описание идеи

Заведем таблицу ST, такую, что она содержит минимумы на всех отрезках, длина которых есть степень двойки.

Имеем $n \log(n)$ интервалов, которые можно вычислить за $n * \log(n)$ с помощью динамического программирования



	2 ⁰	2 ¹	2 ²	2 ³
0	31	31	★	
1	41	41		
2	59	26		
3	26	26		
4	53	53		
5	58	58		
6	97	93		
7	93			

Sparse table

Реализация препроцессинга

Дано: массив A

Таблица $ST[k][i] = \min$ на полуинтервале $[A[i], A[i + 2^k))$.

Формула для вычисления таблицы с помощью динамики:

$$ST[k][i] = \min(ST[k-1][i], ST[k-1][i + 2^{k-1}])).$$

Благодаря ей мы можем сначала посчитать $ST[0]$, $ST[1]$, потом $ST[2]$ и т.д.

[3]	0	0	0							
[2]	3	2	2	2	0	0	0			
[1]	3	6	4	2	2	5	0	0	1	
ST [0]	3	8	6	4	2	5	9	0	7	1
	1	2	3	4	5	6	7	8	9	10
A[i]	3	8	6	4	2	5	9	0	7	1

Sparse table

Реализация препроцессинга

Дано: массив A

Таблица $ST[k][i] = \min$ на полуинтервале $[A[i], A[i + 2^k))$.

Формула для вычисления таблицы с помощью динамики:

$$ST[k][i] = \min(ST[k-1][i], ST[k-1][i + 2^{k-1}])$$

Благодаря ей мы можем сначала посчитать $ST[0]$, $ST[1]$, потом $ST[2]$ и т.д.

[3]	0	0	0																
[2]	3	2	2	2	0	0	0												
[1]	3	6	4	2	2	5	0	0	1										
ST [0]	3	8	6	4	2	5	9	0	7	1									
	1	2	3	4	5	6	7	8	9	10									

input

→

$A[i]$

A[i]	3	8	6	4	2	5	9	0	7	1									
------	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--

Sparse table

Вычисление запроса

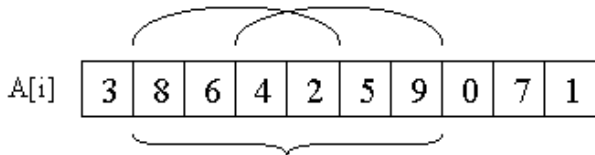
$$RMQ(i, j) = \min(ST[k][i], ST[k][j - 2^k + 1])$$

$$k = \log(j - i + 1)$$

где k - длина максимального двоичного отрезка, который входит в заданный. Двоичные отрезки подбираем так, чтобы один начинался, а второй – заканчивался там, где начинается заданный.

Например,

$$RMQ(1, 6) = \min(ST[2][1] + ST[2][6 - 4 + 1])$$



Sparse table

Оценка сложности

Оценки работы:

Препроцессинг - $O(n \log(n))$,

Запрос - $O(1)$

Sparse table

Ограничения

Какие есть ограничения?

Можно использовать для \min/\max , а можно ли для других функций?


Sparse table


Ограничения


Какие функции подходят?

- min
- max
- наибольший общий делитель
- булевый AND
- булевый OR

Полезные ссылки I

 E-maxx: [sqrt-декомпозиция](https://e-maxx.ru/algorithm/sqrt_decomposition)
https://e-maxx.ru/algorithm/sqrt_decomposition

 Хабр: [Static RMQ](https://habr.com/ru/post/114980/)
<https://habr.com/ru/post/114980/>

 Хабр: [RMQ - 2. Дерево отрезков](https://habr.com/ru/post/115026/)
<https://habr.com/ru/post/115026/>