

Сначала давайте вопросы
насущные

Мячин Данил, ВШЭ ФКН ПМИ

Преисполняемся в структурах данных

Преисполняем в структурах данных

(у нас же их так давно не было)

Преисполняем в структурах данных

(у нас же их так давно не было)

И сразу мини-новость - у нас постепенно возвращаются ДЗ на codeforces'е

Разреженная таблица

Разреженная таблица

Давайте просто представим такие функции, которые удовлетворяют условиям вот таким:

- ассоциативности: $a \circ (b \circ c) = (a \circ b) \circ c$,
- коммутативности: $a \circ b = b \circ a$,
- идемпотентности: $a \circ a = a$.

Разреженная таблица

Давайте просто представим такие функции, которые удовлетворяют условиям вот таким:

- ассоциативности: $a \circ (b \circ c) = (a \circ b) \circ c$,
- коммутативности: $a \circ b = b \circ a$,
- идемпотентности: $a \circ a = a$.

Для нас самый простой пример такой функции (речь идёт про математическую, уточню) - функция минимума чисел, можем чуть-чуть расписать и посмотреть, что для неё эти условия выполняются

Разреженная таблица

Давайте представим, что мы умеем вот что делать:

Для любой позиции i в массиве чисел знаем ответы для отрезков $[i; i]$, $[i; i + 1]$, $[i; i + 3]$, $[i; i + 7]$, $[i; i + 15]$ итд. То есть, мы знаем все отрезки, длина которых является степенью двойки.

Разреженная таблица

Давайте представим, что мы умеем вот что делать:

Для любой позиции i в массиве чисел знаем ответы для отрезков $[i; i]$, $[i; i + 1]$, $[i; i + 3]$, $[i; i + 7]$, $[i; i + 15]$ итд. То есть, мы знаем все отрезки, длина которых является степенью двойки.

А что хотим делать?

Отвечать на любой отрезок

Разреженная таблица

Какой интересный факт есть: предположим нам нужен ответ для отрезка $[i;j]$

Разреженная таблица

Какой интересный факт есть: предположим нам нужен ответ для отрезка $[i;j]$

Тогда, этот отрезок можно разбить на два пересекающихся отрезка длиной 2^k (какая-то степень). Давайте я здесь порисую, чтобы было наглядно понятно

Давайте чтобы быстро находить нужную степень
двойки вот что сделаем

Заведём массив, в который выпишем по возрастанию степени двойки

Давайте чтобы быстро находить нужную степень
двойки вот что сделаем

Заведём массив, в который выпишем по возрастанию степени двойки

Нам достаточно хранить там только те степени, которые меньше или равны
размеру нашего массива

Давайте чтобы быстро находить нужную степень двойки вот что сделаем

Заведём массив, в который выпишем по возрастанию степени двойки

Нам достаточно хранить там только те степени, которые меньше или равны размеру нашего массива

Очевидно, что размер такого массива степеней двоек тогда будет $\log(n)$

Давайте чтобы быстро находить нужную степень двойки вот что сделаем

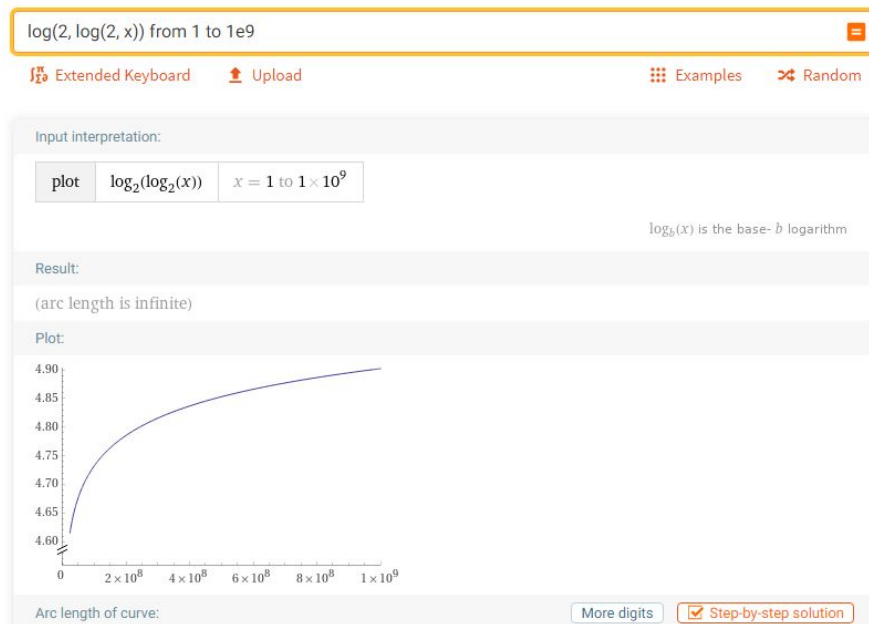
Заведём массив, в который выпишем по возрастанию степени двойки

Нам достаточно хранить там только те степени, которые меньше или равны размеру нашего массива

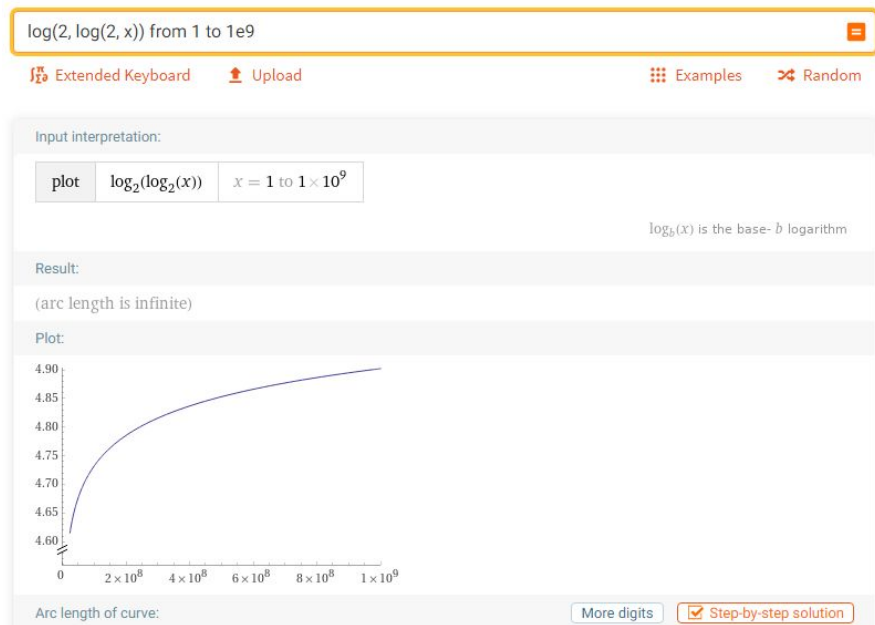
Очевидно, что размер такого массива степеней двоек тогда будет $\log(n)$

Бинарный поиск работает за \log , так тогда если делать бин поиск по такому массиву, тогда относительно n (размера изначального массива) сложность бин поиска будет $\log(\log(n))$

Чтобы вы осознали насколько это маленькое число



Чтобы вы осознали насколько это маленькое число



Для нас - олимпудников, в рамках ограничений на олимпиадах это вообще
чуть ли не константой кажется

Звучит крайне круто, но давайте посмотрим на реализацию

Секрет того, что мы получаем значения для отрезка степени двойки быстро за счёт того, что мы просто будем их все хранить.

Звучит крайне круто, но давайте посмотрим на реализацию

Секрет того, что мы получаем значения для отрезка степени двойки быстро за счёт того, что мы просто будем их все хранить.

Давайте тогда оценим память -

Звучит крайне круто, но давайте посмотрим на реализацию

Секрет того, что мы получаем значения для отрезка степени двойки быстро за счёт того, что мы просто будем их все хранить.

Давайте тогда оценим память - для каждой ячейки будем хранить все отрезки длиной степени двойки

Звучит крайне круто, но давайте посмотрим на реализацию

Секрет того, что мы получаем значения для отрезка степени двойки быстро за счёт того, что мы просто будем их все хранить.

Давайте тогда оценим память - для каждой ячейки будем хранить все отрезки длиной степени двойки

Будет n ячеек с $\log(n)$ значениями, значит суммарно их будет $n \log(n)$

Звучит крайне круто, но давайте посмотрим на реализацию

Секрет того, что мы получаем значения для отрезка степени двойки быстро за счёт того, что мы просто будем их все хранить.

Давайте тогда оценим память - для каждой ячейки будем хранить все отрезки длиной степени двойки

Будет n ячеек с $\log(n)$ значениями, значит суммарно их будет $n \log(n)$

Теперь будем думать про реализацию

Звучит крайне круто, но давайте посмотрим на реализацию

Секрет того, что мы получаем значения для отрезка степени двойки быстро за счёт того, что мы просто будем их все хранить.

Давайте тогда оценим память - для каждой ячейки будем хранить все отрезки длиной степени двойки

Будет n ячеек с $\log(n)$ значениями, значит суммарно их будет $n \log(n)$

Теперь будем думать про реализацию

Очевидная мысль - раз мы используем $n \log(n)$ памяти, быстрее, чем за $n \log(n)$ мы её не заполним

А вот заполнить её за $n \log(n)$ - выполнимая задача

Давайте немного подумаем, для отрезков какой длины в самом начале нам уже всё известно?

А вот заполнить её за $n \log(n)$ - выполнимая задача

Давайте немного подумаем, для отрезков какой длины в самом начале нам уже всё известно?

Да, для отрезков единичной длины уже всё известно - сами элементы

А вот заполнить её за $n \log(n)$ - выполнимая задача

Давайте немного подумаем, для отрезков какой длины в самом начале нам уже всё известно?

Да, для отрезков единичной длины уже всё известно - сами элементы

А сможем ли мы получить отрезок длины два, с помощью двух отрезков длины 1?

А вот заполнить её за $n \log(n)$ - выполнимая задача

Давайте немного подумаем, для отрезков какой длины в самом начале нам уже всё известно?

Да, для отрезков единичной длины уже всё известно - сами элементы

А сможем ли мы получить отрезок длины два, с помощью двух отрезков длины 1?

Если у нас есть два подряд идущих отрезка длины $2^{(i-1)}$, то отрезок какой длины мы можем получить?

А вот заполнить её за $n \log(n)$ - выполнимая задача

Давайте немного подумаем, для отрезков какой длины в самом начале нам уже всё известно?

Да, для отрезков единичной длины уже всё известно - сами элементы

А сможем ли мы получить отрезок длины два, с помощью двух отрезков длины 1?

Если у нас есть два подряд идущих отрезка длины $2^{(i-1)}$, то отрезок какой длины мы можем получить?

Правильно, 2^i

Уже ближе к реализации

Это же таблица! А матрица это что? Самая настоящая таблица! Давайте дадим понимание ij -й ячейке

Уже ближе к реализации

Это же таблица! А матрица это что? Самая настоящая таблица! Давайте дадим понимание ij -й ячейке

(для лучшего понимания, я буду везде писать минимум, но можно и другую функцию, исходный массив чисел назовём x)

Уже ближе к реализации

Это же таблица! А матрица это что? Самая настоящая таблица! Давайте дадим понимание ij -й ячейке

(для лучшего понимания, я буду везде писать минимум, но можно и другую функцию, исходный массив чисел назовём x)

$$st[i][j] = \min(x[i], x[i + 1], \dots, x[i + 2^j - 2], x[i + 2^j - 1])$$

Уже ближе к реализации

Это же таблица! А матрица это что? Самая настоящая таблица! Давайте дадим понимание ij -й ячейке

(для лучшего понимания, я буду везде писать минимум, но можно и другую функцию, исходный массив чисел назовём x)

$$st[i][j] = \min(x[i], x[i + 1], \dots, x[i + 2^j - 2], x[i + 2^j - 1])$$

Как нам вычислять эти значения?

Считаем конкретную ячейку

$$st[i][j] = \min(x[i], x[i + 1], \dots, x[i + 2^j - 2], x[i + 2^j - 1])$$

Считаем конкретную ячейку

$$st[i][j] = \min(x[i], x[i + 1], \dots, x[i + 2^j - 2], x[i + 2^j - 1])$$

Считать каждую заново - было бы слишком жирно. $n \log(n)$ ячеек, каждая считается за n , было бы $n^2 \log(n)$. Больно много. Давайте просто считать ячейки длины 2^j на основе предыдущих по той идее, что была с двумя подряд идущими отрезками длины $2^{(i-1)}$.

Считаем конкретную ячейку

$$st[i][j] = \min(x[i], x[i + 1], \dots, x[i + 2^j - 2], x[i + 2^j - 1])$$

Считать каждую заново - было бы слишком жирно. $n \log(n)$ ячеек, каждая считается за n , было бы $n^2 \log(n)$. Больно много. Давайте просто считать ячейки длины 2^j на основе предыдущих по той идее, что была с двумя подряд идущими отрезками длины $2^{(i-1)}$.

Тогда вычисление одной ячейки занимает единичку

Считаем конкретную ячейку

$$st[i][j] = \min(x[i], x[i + 1], \dots, x[i + 2^j - 2], x[i + 2^j - 1])$$

Считать каждую заново - было бы слишком жирно. $n \log(n)$ ячеек, каждая считается за n , было бы $n^2 \log(n)$. Больно много. Давайте просто считать ячейки длины 2^j на основе предыдущих по той идее, что была с двумя подряд идущими отрезками длины $2^{(i-1)}$.

Тогда вычисление одной ячейки занимает единичку

Тогда саму таблицу можно построить за $n \log n$

Сама реализация

Давайте я с вами напишу её и покажу финтиплюшки, какие можно сделать

Как пока что практиковаться?

Случайно набрёл на астр в раздел курсов и подобного и нашёл задачи на структуры данных

Вот одна из задач на RMQ (минимум на отрезке) -

https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=20&id_topic=44&id_problem=594

Да, тут задача минимума в подматрице, но её можно решить

А также давайте посмотрим, что там есть из того, что мы уже прошли