Алгоритмы 2022-23. Семестр 2.

Задания на практики M3132/33

Большое спасибо @doreshnikov за Latex-шаблон!
По ошибкам и опечаткам обращаться к автору конспекта @kgrigoriy
Версия от 26 марта 2023 г.

План курса:

Неделя 1. Дерево отрезков.

> **Disclaimer.** Под отрезками «от l до r» мы будем почти всегда (если явно не указано иного) подразумевать **полуинтервалы** $[l, r)$, так как с ними удобнее работать.

Напомним задачу, которую мы хотим научиться решать. Дан массив a длины n. Поступают m запросов одного из следующих типов:

a) Вычислить функцию «$\oplus$» на отрезке $[l, r)$: $a_l \oplus a_{l+1} \oplus \ldots \oplus a_{r-1}$ (где $\oplus$ — одна и та же для всех запросов первого типа функция, например, сумма на отрезке);

b) Применить функцию «$\odot\ x$» для i-го элемента: $a_i = a_i \odot x$ (где $\odot$ — одна и та же для всех запросов второго типа функция, например, присвоить элементу новое значение x);

*b**) **[Массовые операции]:** применить функцию «$\odot\ x$» к каждому элементу на отрезке $[l, r)$: $a_i = a_i \odot x$ для всех $i \in [l, r)$ (где $\odot$ — одна и та же для всех запросов второго типа функция, например, присвоить каждому элементу на отрезке новое значение x);

Практика.

В заданиях ниже дан массив a длины n. Требуется придумать, как при помощи дерева отрезков выполнять две операции. Первая операция — присвоить элементу a_i значение x. Вторая операция описана в каждом задании. Обе операции должны работать за $\mathcal{O}(\log n)$.

19. Найти значение суммы $a_l - a_{l+1} + a_{l+2} - \ldots + (-1)^{r-l} a_{r-1}$.

20. Найти значение суммы $a_l + 2a_{l+1} + 3a_{l+2} + \ldots + (r-l)a_{r-1}$.

21. Найти подотрезок $[l_1, r_1)$, такой что $l \le l_1 \le r_1 \le r$ и сумма на подотрезке $[l_1, r_1)$ максимальна среди всех таких отрезков. Достаточно найти значение самой суммы, хотя восстановить отрезок также не составит труда.

22. Найти минимальное i ($1 \le i \le n$), такое что $a_i \ge k$. Здесь k — параметр, который задается в запросе. То есть, в разных запросах значения k могут различаться.

23. Найти минимальное i на отрезке $[l, r)$ ($l \le i < r$), такое что $a_i \ge k$. Здесь k, l и r — параметры, которые задаются в запросе. То есть, в разных запросах значения k могут различаться.

24. Научитесь искать НВП массива длины n за $\mathcal{O}(n \log n)$, используя дерево отрезков. Считайте, что элементы массива — натуральные числа, не превосходящие n.

25. Вычислите количество инверсий в массиве длины n за $\mathcal{O}(n \log n)$, используя дерево отрезков.

26. Дана строка из n открывающих и закрывающих круглых скобок. Придумайте, как при помощи дерева отрезков отвечать на следующие запросы за $\mathcal{O}(\log n)$. Первый запрос — изменить i-ю скобку. Второй запрос — проверить, является ли скобочная последовательность $a_l a_{l+1} \ldots a_r$ правильной.

27. Дана строка из n открывающих и закрывающих круглых скобок. Придумайте, как при помощи дерева отрезков отвечать на следующие запросы за $\mathcal{O}(\log n)$. Первый запрос — изменить i-ю скобку. Второй запрос — найти длину наибольшего префикса отрезка $[l, r)$, который является правильной скобочной последовательностью.

Неделя 2. Дерево отрезков. Массовые операции.

Практика.

В заданиях ниже дан массив a длины n. Требуется придумать, как при помощи дерева отрезков выполнять указанные операции за $\mathcal{O}(\log n)$. Обратите внимание, при выполнении заданий следует уделить особое внимание псевдокоду функций проталкивания отложенных операций `push(...)`.

1. (a) Присвоить значение x всем элементам отрезка
   (b) Найти сумму на отрезке
2. (a) Присвоить значение x всем элементам отрезка
   (b) Умножить все элементы отрезка на -1 (то есть заменить a_i на $-a_i$)
   (c) Найти сумму на отрезке
3. (a) Присвоить значение x всем элементам отрезка
   (b) Умножить все элементы отрезка на -1 (то есть заменить a_i на $-a_i$)
   (c) Найти максимум на отрезке
4. (a) Присвоить значение x всем элементам отрезка
   (b) Умножить все элементы отрезка на -1 (то есть заменить a_i на $-a_i$)
   (c) Найти подотрезок с максимальной суммой
5. (a) Присвоить значение x всем элементам отрезка
   (b) Прибавить значение x ко всем элементам отрезка
   (c) Найти значение элемента
6. (a) Присвоить значение x всем элементам отрезка
   (b) Прибавить значение x ко всем элементам отрезка
   (c) Найти сумму на отрезке
7. (a) Заменить на отрезке a_i на $\max(a_i, x)$
   (b) Заменить на отрезке a_i на $\min(a_i, x)$
   (c) Найти значение элемента
8. (a) Присвоить значение x всем элементам отрезка
   (b) Найти подотрезок максимальной длины, состоящий из одинаковых чисел
9. (a) Прибавить ко элементам отрезка арифметическую прогрессию (то есть к a_l прибавить b, к $a_{l+1} - b + k$, к $a_{l+2} - b + 2k$, и так далее)
   (b) Найти сумму на отрезке

В заданиях ниже дан массив a длины n, состоящий из булевых значений. Требуется придумать, как при помощи дерева отрезков выполнять указанные операции за $\mathcal{O}(\log n)$.

10. (a) Присвоить значение x всем элементам отрезка
    (b) Найти ближайшую к i-му элементу единицу
11. (a) Изменить все значения на отрезке на противоположные
    (b) Найти количество единиц на отрезке
12. (a) Присвоить значение x всем элементам отрезка

   (b) Найти количество непрерывных отрезков из единиц

13. (a) Присвоить значение x всем элементам отрезка
    (b) Найти самый длинный непрерывный отрезок из единиц

14. Даны n прямоугольников на плоскости, стороны которых параллельны осям координат. Найдите точку, покрытую максимальным количеством прямоугольников (если таких точек несколько, можно найти любую). Время работы: $\mathcal{O}(n \log n)$.

15. Даны n прямоугольников на плоскости, стороны которых параллельны осям координат. Найдите площадь той части плоскости, которая покрыта максимальным количеством прямоугольников. Время работы: $\mathcal{O}(n \log n)$.

16. Даны n прямоугольников на плоскости, стороны которых параллельны осям координат, и m точек. Для каждого прямоугольника вычислите, сколько точек он покрывает. Время работы: $\mathcal{O}((n + m) \log(n + m))$.

17. Дан массив длины n. Вычислите количество возрастающих подпоследовательностей массива длины k. Время $\mathcal{O}(kn \log n)$.

18. Есть шахматное поле $n \times n$. Требуется обрабатывать следующие запросы за $\mathcal{O}(\log n)$:
    (a) Добавить ладью на поле
    (b) Удалить ладью с поля
    (c) Найти количество клеток, которые не бьет ни одна ладья

Неделя 3. Sparse Table.

Практика.

28. Как изменится время работы и необходимый объем памяти *Sparse Table*, если хранить значения функции на отрезках длины x^k, а не 2^k $(x > 2)$? Приведите оценки, зависящие от n и x.

29. Дан массив a длины n. Найдите количество отрезков $[l, r)$, таких что $(a_l \text{ or } a_{l+1} \text{ or } \ldots \text{ or } a_{r-1}) > \max(a_l, a_{l+1}, \ldots, a_{r-1})$. Здесь or означает побитовое «ИЛИ». Время работы: $\mathcal{O}(n \log n \log C)$, где C — максимальное из a_i. Подсказка: используйте *Sparse Table*!

30. Дана матрица размера $n \times m$. Постройте структуру, позволяющую отвечать на запросы

    (a) сумма в подматрице $[i_1, i_2] \times [j_1, j_2]$, за время $\langle \mathcal{O}(nm), \mathcal{O}(1) \rangle$

    (b) минимум в подматрице $[i_1, i_2] \times [j_1, j_2]$, за время $\langle \mathcal{O}(nm \log n \log m), \mathcal{O}(1) \rangle$

31. Опишите реализацию разреженной таблицы, поддерживающей следующие операции за время $\langle \mathcal{O}(n \log n), \mathcal{O}(1) \rangle$:

    (a) найти самое правое положительное число на отрезке

    (b) найти самый длинный подотрезок отрезка, состоящий из одинаковых чисел

Неделя 4. Дерево поиска. Декартово дерево поиска.

Практика.

32. Напишите рекурсивную процедуру, выводящую элементы дерева поиска в отсортированном порядке, за время $O(n)$.

33. Напишите нерекурсивную процедуру, выводящую элементы дерева поиска в отсортированном порядке, за время $O(n)$ с $O(1)$ дополнительной памяти (у узлов есть указатели на родителей).

34. Докажите, что нет алгоритма, который строит дерево поиска по заданному массиву из n элементов быстрее, чем за $O(n \log n)$ в худшем случае.

35. Найти в дереве минимальный элемент, не меньший заданного x. Время $O(H)$ (H — высота дерева).

36. Для каждого узла x посчитайте число $w(x)$, равное числу узлов в его поддереве (включая сам x). Время $O(n)$.

37. Используя вычисленные значения $w(x)$, научитесь находить k-й по возрастанию элемент дерева. Время $O(H)$.

38. Используя $w(x)$, научитесь находить по заданному ключу x число элементов, меньших x. Время $O(H)$ (H — высота дерева).

39. По данному узлу дерева найдите следующий в отсортированном порядке за время $O(H)$ с $O(1)$ дополнительной памяти.

40. По данному узлу дерева найдите следующие k узлов в отсортированном порядке за время $O(H + k)$.

41. Приведите пример дерева, в котором средняя глубина узла (среднее расстояние от узла до корня) $O(\log n)$, а высота дерева (максимальное расстояние от узла до корня) — $\omega(\log n)$ (асимптотически больше).

42. Проверить, что заданное дерево является корректным деревом поиска. Время $O(n)$.

43. Дан массив пар x, y, отсортированный по x. Постройте по нему декартово дерево за $O(n)$.

44. Дан массив пар x, y. Все x различны, а y могут совпадать. Проверить, что декартово дерево можно построить единственным образом.

45. Дан массив пар x, y. Все x различны, а y могут совпадать. Найти число способов построить декартово дерево.

46. Пусть у дерева поиска нет узлов с одним ребенком (то есть у каждой внутренней вершины ровно два ребенка). Правда ли, что высота такого дерева $O(\log n)$?

47. Петя решил соптимизировать декартово дерево по памяти, не храня ключи y. Он рассуждает так: все ключи y — случайные числа, если мы сравним два независимых случайных числа y_1 и y_2, мы получим результат больше или меньше с вероятностью 50%. Давайте просто в тех местах кода, где производится сравнение y, выберем случайный вариант с вероятностью 50%. Покажите, что после такого изменения некоторая последовательность действий может привести к тому, что высота дерева (вернее, ее матожидание) будет $\Omega(n)$. Почему так вышло?

48. Как можно починить метод Пети, если для каждой вершины посчитан размер поддерева?

49. В дереве поиска, в отличие от дерева отрезков, исходные элементы множества хранятся не только в листьях, но и в промежуточных узлах. Иногда это неудобно, поэтому делают другую версию дерева поиска: исходные элементы множества хранятся только в листьях, а в промежуточных вершинах хранится максимальный ключ в поддереве. Покажите, как осуществлять операции поиска и добавления элемента в таком дереве.

50. Покажите, что структура дерева из предыдущего задания изоморфна обычному дереву поиска на $n-1$ элементах, а значит его можно балансировать всеми теми же способами.