

Реализация запроса в дереве отрезков сверху

Данная операция позволяет выполнять запросы на дереве отрезков, причем алгоритм запускается от корня и рекурсивно идет сверху вниз.

Содержание

- 1 Алгоритм
- 2 Пример
- 3 Реализация
- 4 См. также
- 5 Источники информации

Алгоритм

Замечание. Используем в алгоритме не отрезки, а полуинтервалы (левая граница включительно, а правая — нет).

Пусть есть уже построенное дерево отрезков и идет запрос на полуинтервале $[a \dots b)$.

В качестве параметров рекурсий передаем следующие переменные:

- *node* — номер (в массиве с деревом отрезков) текущей вершины дерева.
- *a, b* — левая и правая границы запрашиваемого полуинтервала.

Пусть *l, r* — это левая и правая границы полуинтервала, за которые "отвечает" наша вершина. Запустим рекурсивную процедуру от всего полуинтервала (то есть от корневой вершины).

Для текущего состояния проверяем следующие условия :

- Если текущий полуинтервал не пересекается с искомым, то возвращаем нейтральный элемент.

Например: текущий $[1 \dots 3)$, а искомый $[3 \dots 5)$;

- Если текущий полуинтервал лежит внутри запрашиваемого полуинтервала, то возвращаем значение в текущей вершине.

Например: текущий $[2 \dots 3)$, а искомый $[2 \dots 4)$;

- Иначе переходим к рекурсивным вызовам функций от детей вершины. При этом возвращаем значение на текущем полуинтервале, как функцию (соответствующую типу нашего запроса) от результатов выполнения на детях.

Так как на каждом уровне дерева рекурсия может прийти до не более, чем двух вершин (иначе бы нашлось две рядом стоящие вершины одного уровня, объединение которых дало отрезок, за который отвечает вершина предыдущего уровня), а всего уровней $\log n$, то операция выполняется за $O(\log n)$.

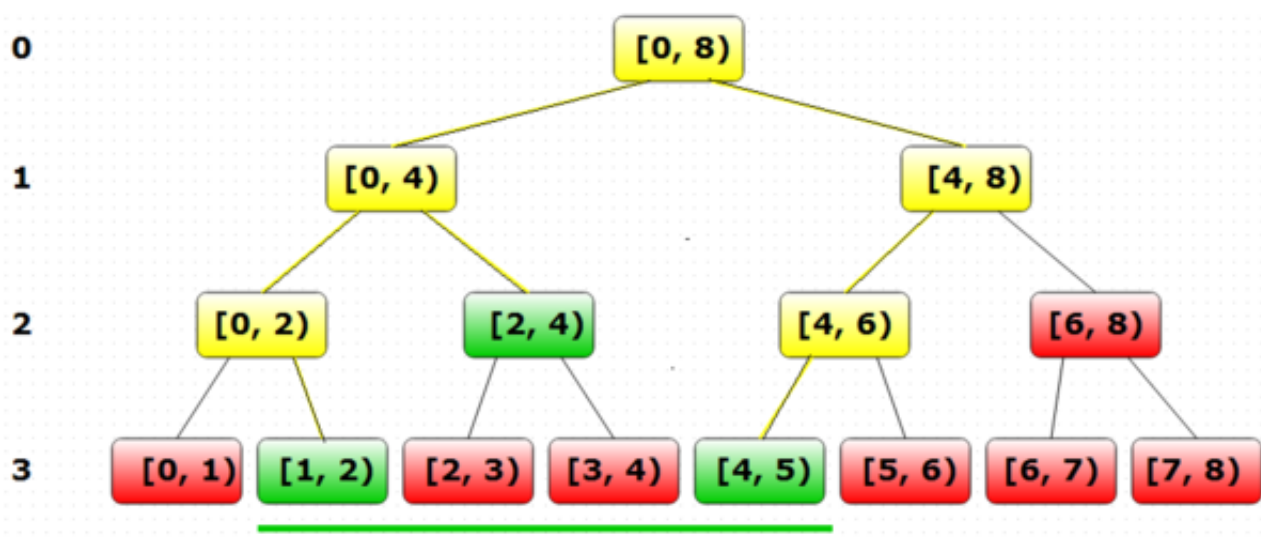
Пример

Рассмотрим данный алгоритм на примере задачи RSQ (Range Sum Query — запрос суммы на отрезке).

При этом сумма на текущем полуинтервале (в случае вызова рекурсий от детей) равна сумме результатов выполнения операции на этих детях.

Пусть дерево содержит 8 листьев и запрашиваемая сумма — это отрезок $[1 \dots 4]$ (полуинтервал $[1 \dots 5)$).

Рассмотрим
данный
алгоритм на



определенных глубинах рекурсии, то есть на разных уровнях дерева (на рисунке глубина обозначена слева от уровня):

- На глубине 0.
 - Текущий полуинтервал $[0 \dots 8)$ пересекается с $[1 \dots 5) \Rightarrow$ рекурсивно переходим к $[0 \dots 4)$ и $[4 \dots 8)$
- На глубине 1.
 - $[0 \dots 4)$ и $[4 \dots 8)$ пересекаются с $[1 \dots 5) \Rightarrow$ переходим по рекурсивным вызовам на полуинтервалах $[0 \dots 2)$, $[2 \dots 4)$, $[4 \dots 6)$ и $[6 \dots 8)$
- На глубине 2.
 - $[0 \dots 2)$ и $[4 \dots 6)$ пересекаются с $[1 \dots 5) \Rightarrow$ переходим в листья $[0 \dots 1)$, $[1 \dots 2)$, $[4 \dots 5)$, $[5 \dots 6)$

- $[2 \dots 4)$ полностью лежит внутри $[1 \dots 5)$ \Rightarrow возвращаем сумму на этом отрезке
- $[6 \dots 8)$ не пересекается с $[1 \dots 5)$ \Rightarrow возвращаем нулевое значение

- На глубине 3.

- Листья $[1 \dots 2)$, $[4 \dots 5)$ лежат в запрашиваемом интервале \Rightarrow возвращаем значения в них
 - Листья $[0 \dots 1)$, $[5 \dots 6)$ лежат вне $[1 \dots 5)$ \Rightarrow возвращаем нейтральное значение

Таким образом ответ на полуинтервале $[1 \dots 5)$ равен сумме значений в вершинах, отвечающих за полуинтервалы $[1 \dots 2)$, $[2 \dots 4)$ и $[4 \dots 5)$.

Реализация

Рассмотрим реализацию задачи о дереве отрезков с произвольной ассоциативной бинарной операцией.

Пусть в узлах дерева хранятся структуры из трех полей:

- `left` — левая граница полуинтервала, за который "отвечает" текущая вершина.
- `right` — правая граница этого полуинтервала.
- `res` — результат операции на полуинтервале.

```
int query(int node, int a, int b)
{
    l = tree[node].left
    r = tree[node].right
    if [l, r) ∩ [a, b) == ∅
        return ε // ε — нейтральный для данной операции элемент
    if [l, r) ⊂ [a, b)
        return tree[node].res
    return query(node * 2 + 1, a, b) ◦ query(node * 2 + 2, a, b)
}
```

См. также

- Реализация запроса в дереве отрезков снизу
- Дерево отрезков. Построение

Источники информации

- MAXimal :: algo :: Дерево отрезков (http://e-maxx.ru/algorithm/segment_tree)
- Википедия — Дерево отрезков (http://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_%D0%BE%D1%82%D1%80%D0%B5%D0%B7%D0%BA%D0%BE%D0%B2)
- Дискретная математика: Алгоритмы — Визуализатор дерева отрезков (<http://rain.ifmo.ru/cat/view.php/vis/trees/segment-2006>)

Источник — «http://neerc.ifmo.ru/wiki/index.php?title=Реализация_запроса_в_дереве_отрезков_сверху

title=Реализация_запроса_в_дереве_отрезков_сверху&oldid=65295»

- Эта страница последний раз была отредактирована 10 мая 2018 в 14:30.