Дерево отрезков. Введение

Пусть задана некоторая упорядоченная последовательность элементов $A = \{a_0, a_1, \ldots, a_{n-1}\}$ Дерево отрезков применяется для решения ряда задач, связанных с вычислением некоторои ассоциативной операции $\circ$ на произвольном интервале $[b; e)$ элементов последовательности $S(b, e) = a_b \circ a_{b+1} \circ \ldots \circ a_{e-1}$.

Идея заключается в том, что помимо элементов $a_i = S(i, i+1)$ хранятся также частичные суммы вида $S(i2^k, (i+1)2^k)$ для всех $k \in [1; \log_2 n]$. В итоге при нахождении произвольной $S(b, e)$ можно использовать заранее найденные суммы вида $S(i2^k, (i+1)2^k)$

Свойства дерева отрезков

Корень интервала

Назовем некоторое число $R(b, e)$ корнем интервала $[b; e)$ если выполняются следующие свойства:

- $R(b, e) \in [b; e]$ (то, что интервал является закрытым не опечатка);
- $R(b, e)$ делится без остатка на некоторую степень двойки 2^k;
- Все элементы интевала $[b; e]$ не делятся на 2^{k+1}.

Предложение 1.

Для любого непустого интервала $[b; e)$ существует единственный корень $R(b, e)$
Доказательство. Если интервал не пустой, то очевидно в нем найдется корень. Докажем от противного, что он единственный.

Пусть корень не единственный и в интервале найдутся два корня вида $p2^k$ и $q2^k$. Для определенности $p < q$. Тогда $p + 1 \leq q$ и $(p+1)2^k \in [b; e]$ Из чисел p и $p + 1$ одно является четным, следовательно одно из чисел $p2^k$ и $(p+1)2^k$ делится на 2^{k+1}, что противоречит свойству корня.

Предложение 2.

Пусть корень интервала $[b; e)$ совпадает с началом интервала b. Тогда интервал $[b; e)$ можно представить в виде объединения $h(b, e)$ интервалов вида $[i2^r; (i+1)2^r)$, где $h(b, e)$ — количество единиц в двоичной записи числа $e - b$
Доказательство. Запишем последовательность чисел $e = e_1, e_2, e_3, \ldots e_m = b$, в которой каждое очередное число e_{j+1} получается из e_j заменой самой правой единицы в двоичной записи числа на ноль. Если e_{j+1} отличается от e_j в разряде r, то $e_j = e_{j+1} + 2^r$ и e_j можно представить в виде $i2^r$, Таким образом, интервал $[e_{j+1}; e_j)$ имеет вид $[i2^r; (i+1)2^r)$, а количество таких интервалов совпадает с количеством единиц в двоичной записи $e - b$

Пример 1.

Рассмотрим представление интервала $[160; 179)$

$$\begin{array}{lll} e_1 & (10110011) & 179 \\ e_2 & (10110010) & 178 \\ e_3 & (10110000) & 176 \\ e_3 & (10100000) & 160 \end{array}$$

Предложение 3.

Пусть корень интервала $[b; e)$ совпадает с концом интервала e и имеет вид $t2^k$. Тогда интервал $[b; e)$ можно представить в виде объединения $v(b, k)$ интервалов вида $[i2^r; (i+1)2^r)$, где $v(b, k)$ — количество нулей в младших разрядах двоичной записи числа b, находящихся слева от самой младшей единицы..

Доказательство. Запишем последовательность чисел $b = b_1, b_2, b_3, \dots b_m = e$, в которой каждое очередное число $b_{j+1} = b_j + 2^r$, где r это позиция самой младшей единицы в двоичной записи числа b_j. Тогда b_j можно представить в виде $i2^r$, а интервал $[b_j; b_{j+1})$ имеет вид $[i2^r; (i+1)2^r)$,

Пример 2.

Рассмотрим представление интервала $[141; 160)$

$$\begin{array}{lll} b_1 & (10001101) & 141 \\ e_2 & (10001110) & 142 \\ e_3 & (10010000) & 144 \\ e_3 & (10100000) & 160 \end{array}$$

Теорема

Всякий интервал $[b; e)$ может быть представлен в виде объединения не более чем $2\log_2 n$ интервалов вида $[i2^r; (i+1)2^r)$.

Хранение частичных сумм

Для хранения элементов последовательности $A = \{a_0, a_1, \dots, a_{n-1}\}$ и всех частичных сумм вида $S(i2^k, (i+1)2^k)$ можно использовать вектор из $2n$ элементов. При этом частичная сумма $S(i2^k, (i+1)2^k)$ должна храниться в элементе с номером $(2i+1)2^k$.

Очевидно, что всякое натуральное число обладает единственным представлением в виде $(2i+1)2^k$, поэтому все частичные суммы будут храниться в элементах с разными номерами.

Реализация

Пусть заданы границы интервала b и e. *binop* обозначает некоторую бинарную ассоциативную операцию, *one* — нейтральный элемент относительно этой операции.

Первый цикл удаляет единицы из правой границы интервала, пока не будет достигнут корень этого интервала. Проверка `e>0` требуется, чтобы избежать зацикливания, когда `b=0`. Выражение `e&(e-1)` убирает самую правую единицу в двоичной записи числа.

Второй цикл находит разложение на отрезки для интервала от левой границы до корня.

```
T R=one,L=one;
for (unsigned t=e&(e-1);t>=b && e>0; e=t,t=e&(e-1))
    R=binop(S[t+e],R);
for (unsigned t=(b|(b-1))+1;t<=e;b=t,t=(b|(b-1))+1)
    L=binop(L,S[b+t]);
return binop(L,R)
```

```
S[2*a+1]=s;
for (unsigned p=1;p<n;p<<=1)
   a&=~p,S[2*a+2*p]=binop(S[2*a+p],S[2*a+3*p]);
```