Задачи RMQ и LCA

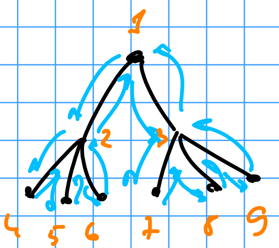
Сведение:

RMQ → LCA (Декартово дерево)

LCA → RMQ (Эйлеров обход)

⇒ Можем решить за $(O(n \log n), O(1))$

NB: при сведении LCA к RMQ мы получаем ±1-RMQ



→ (1, 0) (2, 1) (4, 2) (2, 1) (5, 2) . . . .

вершина   глубина

!!! RMQ по глубине !!!

⇒ Значения для RMQ на ∀ шаге изменяются либо на +1, либо на −1

±1-RMQ

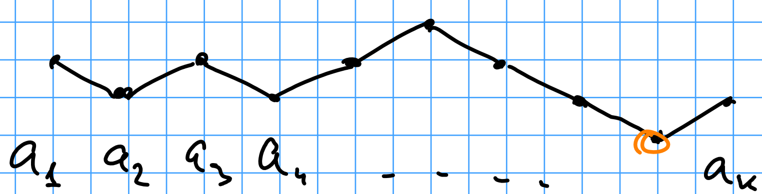
Вход задачи: $a_1, a_2, a_3, \ldots, a_n$ — числа;

$$|a_i - a_{i+1}| = 1$$

Разобьём $a_1 \ldots a_n$ на n/k блоков $A_1 \ldots A_{n/k}$

$A_1 = (a_1, a_2, a_3, \ldots, a_k)$

$A_2 = \ldots$



∀ блока найдём min.

Разных типов блоков может быть $\leq 2^{k-1}$

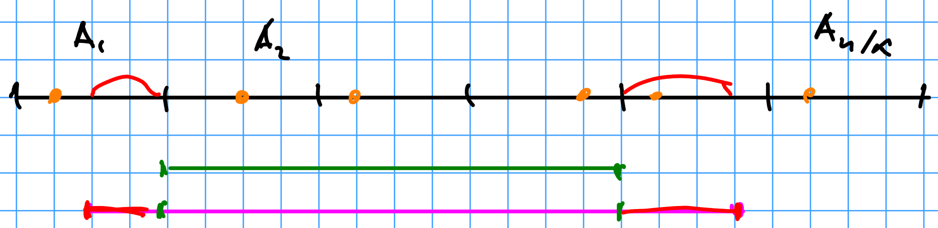
Построим табличку: тип блока $\longrightarrow$ индекс min

Расширим эту табличку всеми начальными и конечными отрезками $\forall$ блока.

Место $\underline{\underline{2^k \cdot k}}$

Для блоков $A_1 \dots A_{n/k}$ вычислим минимумы $m_1 \dots m_{n/k}$ и для них построим разреж. таблицу.

$$\left(O\left(\frac{n}{k}\cdot\log\frac{n}{k}\right),\ O(1)\right)$$



Если запрос "крайен" блокам $\Rightarrow$ min за $O(1)$ по разр. таблице.

Если не "крайен", то кусочки по краям вычислим исходя из типа блока за

$$\underbrace{O(1)}_{\text{разр. табл.}} + \underbrace{O(1)}_{\text{таблица профилей}}$$

$$O\Big(\underbrace{2^k\cdot k}_{\text{профили}} + \underbrace{\frac{n}{k}\log\frac{n}{k}}_{\text{разр. табл.}} + n\Big) \quad]\ k = \frac{1}{2}\log n$$

$$O\Big(\underbrace{n^{1/2}\cdot\frac{1}{2}\log n}_{o(n)} + \underbrace{\frac{n}{\frac{1}{2}\log n}}_{O(1)}\cdot\underbrace{\log(n/\tfrac{1}{2}\log n)}_{O(n)} + n\Big) = O(n)$$

$\Rightarrow$ LCA и RMQ за $(O(n), O(1))$ (статическая постановка)

Числовые алгоритмы

Число N можно записать $n = \log_2 N$ битов

Сложение:

За $O(n)$

Умножение:

$O(n^2)$ $\quad O(n^{\log_2 3})$ $\quad O(n \log n)$

Карацуба $\quad$ БПФ

```
Mult(a, b):
    if b == 0:
        return 0
    r = Mult(a, ⌊b/2⌋)
    if b mod 2 == 1:
        return 2·r + a
    else return 2·r
```

$O(n^2)$

123	15
61	30
30	60
15	120
7	240
3	480
1	960

$\Sigma = 123 \cdot 15$

Деление:

```
Div(a, b):
    if a < b:
        return (0, a)
    (q, r) = Div(⌊a/2⌋, b)
    q = 2·q
    r = 2·r
    if a mod 2 == 1:
        r = r + 1
    if r ≥ b:
        r = r - b
        q = q + 1
    return (q, r)
```

$O(n^2)$

```
Div(123, 15)
    (4, 1) = Div(61, 15)
    q = 4·2
    r = 1·2
    if 123 mod 2 == 1:
        r = 2 + 1
```

Возведение в степень:

```
Pow(a, k):
  if k == 0:
    return 1
  y = Pow(a, ⌊k/2⌋)
  if k mod 2 == 1:
    return y^2 · a
  return y^2
```

$|a| = n$

$O(\log k \cdot (n \cdot k)^2)$

$O(n^3 \cdot 2^{2n})$

Модульная арифметика

4 операции в кольце mod M

$\log M = n$

Сложение: $O(n)$

$a + b \mod M$

```
if a + b > M
    return a + b - M
return a + b
```

Умножение:

$a \cdot b \mod M = \underbrace{\mathrm{Div}}_{O(n^2)}(\underbrace{\mathrm{Mult}(a, b)}_{O(n^2)}, M)$ $\quad O(n^2)$

Возведение в степень:

$a^b \mod M = \mathrm{POW}(a, b)$ $\quad O(n^3)$

(умножаем mod M)

Деление:

$$a/b \mod M \iff a\cdot b^{-1} \mod M$$

- b^{-1} определён $\iff$ НОД$(b, M) = 1$

GCD – Greatest Common Divisor

HCF – Highest Common Factor

Алгоритм Евклида

$$\text{НОД}(a, b) = \text{НОД}(b, a \bmod b)$$

```
GCD(a, b):
  if b == 0:
    return a
  return GCD(b, a mod b)
```

Расширенный алгоритм Евклида

Extended Euclid's Algorithm

$$\text{НОД}(a, b) = \text{НОД}(b, a \bmod b) = d$$

$$a = bq + r$$
$$b = rq' + r'$$

$$d = ax + by =$$
$$= bx' + (a \bmod b)\cdot y' =$$
$$= bx' + (a - \underbrace{\lfloor \tfrac{a}{b} \rfloor}_{q'}\cdot b)\cdot y'$$
$$= \underbrace{ay'}_{x} + b(\underbrace{x' - \lfloor \tfrac{a}{b} \rfloor y'}_{y})$$

```
EGCD(a, b)
  if b == 0:
               d  x  y
    return (a, 1, 0)
  (d, x', y') = EGCD(b, a mod b)
  return (d, y', x' - ⌊a/b⌋y')
```

обычное деление

Вычисление обратного

Найти b^{-1}?

НОД$(M, b) = 1 \iff \exists\, x, y:$

$$\cancel{Mx} + b \cdot \underbrace{y}_{=b^{-1}} = 1 \mod M$$

$b^{-1} = y$ из EGCD(M, b).

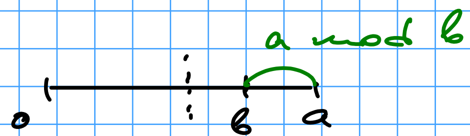
Время работы EGCD

Утв: Если $a > b$, то $\underline{a \bmod b} \le a/2$

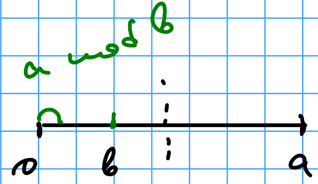
- $b > a/2$

$a \bmod b =$

$= a - b < a/2$



- $b \le a/2$



$a \bmod b < b \le a/2$

$\Rightarrow$ $\forall$ вызов GCD уменьшает один из аргументов вдвое. $\Rightarrow$ EGCD работает $O(n \cdot n^2) = O(n^3)$

$\Rightarrow$ Деление mod M тоже работает за $\underline{O(n^3)}$.

Числа Фибоначчи по модулю

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \qquad A^N = \begin{pmatrix} F_n & F_{n-1} \\ F_{n-1} & F_{n-2} \end{pmatrix}$$

$$A^{n-1} = \begin{pmatrix} F_{n-1} & F_{n-2} \\ F_{n-2} & F_{n-3} \end{pmatrix}$$

$$A^{n-1} \cdot A = \begin{pmatrix} \underbrace{F_{n-1} + F_{n-2}}_{} {}^{F_n} & F_{n-1} \\ F_{n-1} & F_{n-2} \end{pmatrix} \qquad O(n^2 \cdot \underbrace{\log N}_{n})$$

$$O(n^3)$$