1) Находить сумму за $O(\log n)$
2) Увеличить знач-е элемента на Δ за $O(\log n)$

$f(x) = x \,\&\, (x+1)$ — обнуление НБПЕ
$g(x) = x \mid (x+1)$ — увеличение НБПЕ хотя бы на 1 единицу.

Утв. $\forall x \in \mathbb{Z}^+ \hookrightarrow f(g(x)) \le f(x)$

Опр. Пусть $\varphi: X \to X$. $\varphi^n(x) := \underbrace{\varphi(\varphi(\varphi \dots \varphi(x)))\dots)}_{n \text{ ур-й влож-ти}}$
$\varphi^0(x) := x$

Опр. $\tilde{f}(x) = f(x) - 1$

Утв. $f(\tilde{f}^n(x)) = 0$, $n = O(\log x)$

$\square$ Покажем, что $f(x)$, $f(f(x)-1)$, ... убывает.
Дост. док-ть, что $f(x) > f(f(x)-1)$
Пусть $x = \overline{x_1 x_2 x_3 \dots x_n}$ и НБПЕ нач-ся с p. Тогда $f(x) = \overline{x_1 x_2 \dots x_{p-1} \underbrace{0 \dots 0}_{n-p+1}}$

Когда мы вычтем из $f(x)$ единицу то цифры с $p-1$ по n точно 1.
$\Rightarrow$ из $f(f(x)-1)$ обнулится хотя бы $n-p+2$ цифр, а в $f(x)$ обнулится $n-p+1$.
$\Rightarrow f(f(x)-1) < f(x)$.
Заметим, что мы можем применять $\tilde{f}(x)$, пока в x есть биты. Заметим, что их число $O(\log x)$ $\blacksquare$

Итоговая конструкция

$$F_A[m] = \sum_{k=f(m)}^{m} A[k]$$

$$\sum_{k=0}^{m} A[k] = \sum_{k=f(m)}^{m} A[k] + \sum_{k=f(f(m)-1)}^{f(m)-1} A[k] + \dots =$$

$$= F_A[m] + F_A[f(m)-1] + \dots$$

```
def sum(m):
  sum = 0
  for k = m, k > 0, k = k&(k+1)-1:
    sum = sum + F_A[k]
```

$O(\log n)$ см. на k.

Обновление

Утв. Необх. и дост-но обновить суммы, для которых верно $f(m) \le k \le m$.

$\square$ Мы суммируем от $f(m)$ до m. $\blacksquare$

Утв. Нужно обновить все такие ячейки F_A с индексами которые равны $g^n(k)$, $n \in \mathbb{Z}^+$.

$\square$ $g^0(k) = k$. Очевидно, обновляем.

Вспомним, что $f(g(k)) \le f(k)$, а также $g(k) > k$. Тогда:

$$\dots \le f(g(k)) \le f(k) \le k < g^1(k) < g^2(k) < \dots$$

По утв. выше получаем, что надо обновить $g(k)$. Рассуждая по индукции получаем, что обновить надо и $g^2(k)$....

Покажем, что др. эл-ты обн-ть не надо. Пусть $\exists m: \ m \ne g^t(k), \ t \in \mathbb{Z}^+$, но требует обн-я.

Пусть размер НБПЕ m равен p. Занулим p самых правых цифр у k и обозначим k_0.

$m = \overline{m_1 m_2 m_3 .. m_n}$, $K = \overline{K_1 K_2 \ldots K_L}$, если:

1) $K = f(m)$: $K_0 = f(m) = K$.

2) $K > f(m)$:, то $\exists i$: $K_i > f(m)_i$, т.е. $K_i = 1$, $f(m)_i = 0$. Причем ост. цифры левее i между K и $f(m)$ равны.

Рассмотрим случаи.

1) K_i нах-ся в самых правых p символах. Тогда произойдет обнуление этого числа, префиксы $f(m)$ и k равны $\Rightarrow f(m) = K_0$.

2) Если K_i не нах-ся в самых p правых символах, $K_0 > f(m)$

$\Rightarrow f(m) \le K_0$. Пусть $f(m) < K_0$. $\Rightarrow$ в числ. K есть такой старший бит $- 1$, а в m он равен нулю. Но тогда $K > m$. Противоречие.

$\Rightarrow f(m) = K_0$. $K_0 = \overline{K_1 K_2 \ldots \underbrace{000..0}_{p}}$

Заполним блок нулей единицами применяя g к K_0.

$\Rightarrow g^p(K_0) = g^t(K)$ (т.к. кол-во единиц в K может быть меньше, чем в $g^p(K_0)$).

Но $g^p(K_0)$ - то что такое?

K_0 - K с обнул. хвостом длины p.

$f(m)$ - m с обнул. хвостом длины p.

Но т.к. $f(m) = K_0$, то $m = g^p(K_0)$

Противоречие ∎

Условия на операцию

Пусть есть операция
$\oplus : X \times X \to X$

1) $\oplus$ – ассоциативна:
$\forall a, b, c \in X \hookrightarrow\ a \oplus (b \oplus c) =$
$= (a \oplus b) \oplus c.$

Этим пользуемся, считая сумму на префиксе

```
def add(K, Δ):
    For K' = K, K' < |F_A|, K' = K' | (K'+1):
        F_A[K'] = F_A[K'] + Δ.
```

$O(\log n)$! КАЖДОЕ ВЗЯТИЕ g МЫ УСТАНАВЛИВАЕМ ХОТЯ БЫ ОДИН БИТ В ЕДИНИЦУ, ВСЕГО ЭТО МОЖНО СДЕЛАТЬ, СКОЛЬКО БИТ В ЧИСЛЕ.

2) $\exists \ominus : X \times X \to X :$
$(a \oplus b) \ominus b = a$

Исп-м, считая сумму на отрезке.

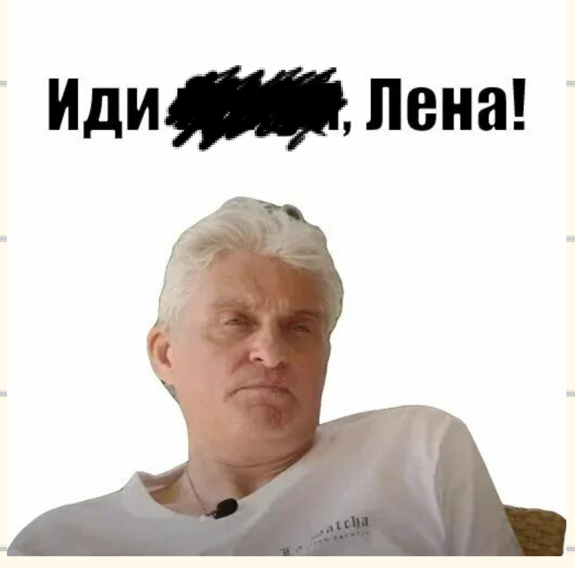
3) Коммут-ть. $\forall a, b \in X: \ a \oplus b = b \oplus a$
исп-м, когда умн-м на Δ.

Построение $O(n)$

#здесь в преф. сумму добавляем A[m]!

- Иници-м F_A как массив преф. сумм.
- Идем с конца: $F_A[m] = F_A[m] - F_A[F(m)-1]$

Усё!



Двумерное дерево Фенвика

Пусть A массив $n \times m$. Хотим (нет):

1) Находить сумму в прямоуг-ке за $O(\log n \cdot \log m)$
2) Изм. знач-е эл-та за $O(\log n \cdot \log m)$

Def. Двумерным деревом Фенвика массива A $n \times m$ назовем массив $n \times m$ F_A, где:

$$F_A[a][b] = \sum_{k=F(a)}^{a} \sum_{t=F(b)}^{b} A[k][t]$$

Утв. Для отв-а в т. необх-мо и дост-но учитывать сумма, для которых верно:

$F(a) \le x \le a$

$F(b) \le y \le b$

```
def sum(a, b):
    sum = 0
    for k = a, k > 0, k = F(k)-1:
        for t = b, t > 0, t = F(t)-1:
            sum = sum + F_A[k][t]

    return sum
```

Работает за $O(\log n \log m)$

Утв. Нужно обновить все такие ячейки F_A с индексами, которые равны $(g^n(a), g^m(b))$; $n, m \in \mathbb{Z}^+$

```
def add(a, b, Δ):
    for k = a, k < n, k = k | (k+1):
        for t = b, t < m, t = t | (t+1):
            F_A[k][t] = F_A[k][t] + Δ
```