Задачи RMQ и LCA

≡ RMQ = Range Minimum Query

Input: Array $[1, N]$

Query $(i, j) \longrightarrow \min A[i, j]$

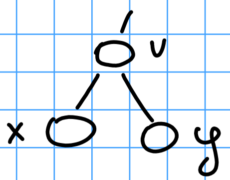
$O(n)$, "прямое вычисление"

Change $(i, x) \Longrightarrow A[i] \leftarrow x$

Динамическая постановка RMQ.

Дерево отрезков

Корневое дерево, где # вершине соотв. отрезок



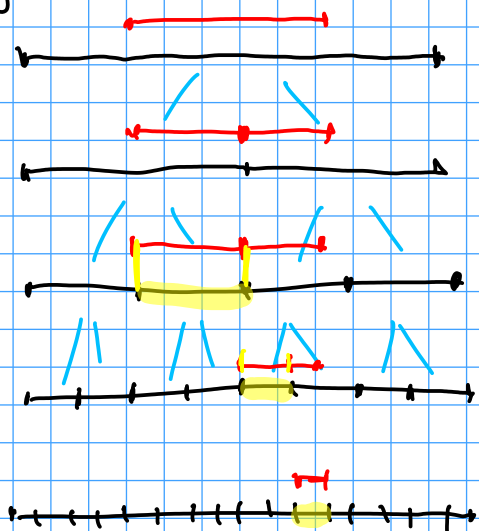
$\Delta(v)$ – отрезок соотв. v, $[i, j]$

$\Delta(x) = [i, m]$

$\Delta(y) = [m+1, j]$

$m = \lfloor (i+j)/2 \rfloor$

Корню соотв $[1, N]$



```
Decompose (Δ, v):
   if Δ = Δ(v) ≠ ∅
       return Δ(v)
   if Δ ∩ Δ(left(v)) ≠ ∅
       S₁ = Decompose (Δ ∩ Δ(left(v)), left(v))
   if Δ ∩ Δ(right(v)) ≠ ∅
       S₂ = Decompose (Δ ∩ Δ(right(v)), right(v))
   return S₁ ∪ S₂
```

Утв 1: Получен. отрезки не пересек.

Утв 2: $\bigcup_i \Delta_i = \Delta$, где $\{\Delta_i\}$ = Decompose(Δ, root)

Query(i,j):

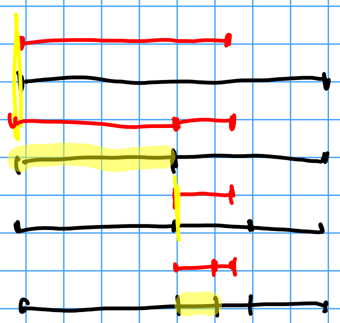
$\{\Delta_i\} \leftarrow$ Decompose((i,j), root)
return $\min \{\Delta_i\}$

Change(i, x):

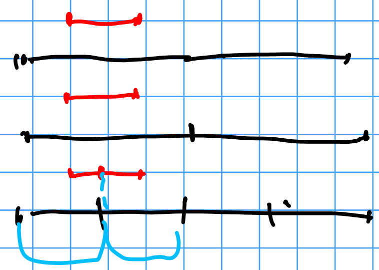
$v_1 \dots v_k \leftarrow$ путь бинарного поиска
update(v_i)

Утв. 3:

Decompose возвр. $O(\log n)$ отрезков.



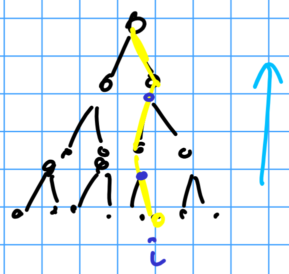
$\Rightarrow O(\log n)$



Утв 4.

Query(i,j) требует $O(\log n)$ операций

Утв. 5: Можно посчитать min для всех отр. дерева за $O(n)$



Утв 6: При Change(i, x) нужно обновить $O(\log n)$ отрезков и это можно сделать $O(\log n)$ операций.

NB: – можно хранить в массиве
– Подходит для $\forall$ ассоц. операции например, RSQ
...

Статическая постановка RMQ.

Сложность $(f(n), g(n))$

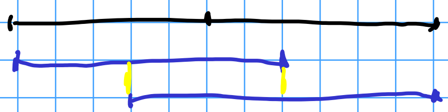
$f(n)$ – сложность предобработки

$g(n)$ – сложность запроса

Прямое вычисление $(O(1), O(n))$

Полное предвычисление $(O(n^2), O(1))$

Утв: $\forall$ отрезок раскладывается на 2 пересек. отрезк., длина которых – степень двойки, либо он сам длины степень двойки



$\min(a, a) = a$

„Разреженная таблица" $(O(n \log n), O(1))$

NB: RSQ

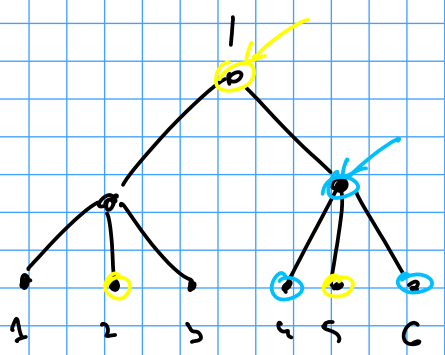
$(O(n), O(1))$

$$P[k] = \sum_{i \le k} A[i]$$

$$Query(i, j) = P[j] - P[i-1]$$

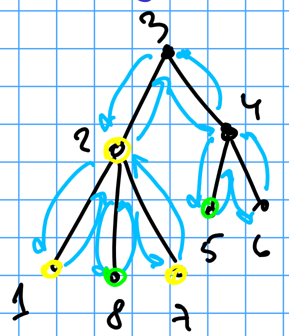
Задача LCA

Least Common Ancestor



$LCA(i, j) \to$ первого общ. предка

Сведение LCA к RMQ точнее к ±1 RMQ



Эйлеров обход (длина $2N$)

3 2 1 2 8 2 7 2 3 4 5 4 6 4 3

↓

$(3,0), (2,1), (1,2), (2,1), (8,2), \dots$

Утв: $LCA(i,j) = RMQ(F(i), F(j))$ по глубине

$F(i)$ – первое упоминание вершины i

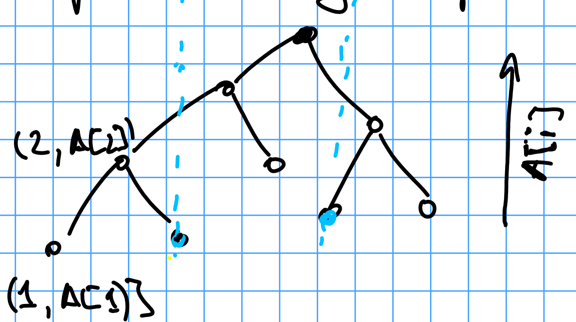
Следствие:

$(O(n \log n), O(1))$

Сведение RMQ к LCA

$A[1,N] \longrightarrow (1, A[1]), (2, A[2]), (3, A[3]) \dots$

Построим декартово дерево за $O(N)$



$RMQ(i,j) = LCA(i,j)$

$RMQ \to LCA \to RMQ$

±1-RMQ

$$\begin{matrix} & {\scriptstyle +1} & & {\scriptstyle +1} & & {\scriptstyle -1} & & {\scriptstyle +1} & & {\scriptstyle +1} & & {\scriptstyle -1} & & {\scriptstyle -1} & \\ 0 & & 1 & & 2 & & 1 & & 2 & & 3 & & 2 & & 1 & & 0 \end{matrix}$$

RMQ за $(O(n), O(1))$

На вход подали $a_1, a_2, \dots, a_n$

Разобьём на блоки по k эл-ов

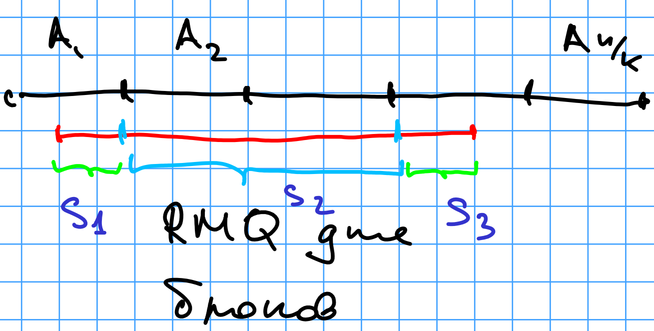
$A_1, A_2, \dots, A_{n/k}$ – блоки

$A_1 = (a_1, a_2, \dots, a_k)$

$\vdots$

Для $\forall$ блока посчитаем min

и построим разр. табл. для $A_1, \dots, A_{n/k}$



$\forall$ блок $\quad 1 \; 2 \; 3 \; 4 \; 3 \; 4 \; 3 \; 2 \; 3$

$\downarrow$

$1 \; \underbrace{+1 \; +1 \; +1 \; -1 \; +1 \; -1 \; -1 \; +1}_{\text{профиль}}$

Всего 2^{k-1} различн. профилей.

Для $\forall$ профиля знаем, где у него минимум.

$S \to S_1 \cup S_2 \cup S_3$

Для S_2 вычисляем RMQ для блоков
Для S_1 и S_3 определяем тип профиля
и находим min эл-т по профилю.

Запрос: $O(1)$

Предобработка: $O\left(\frac{n}{k}\log\frac{n}{k} + n + \underbrace{2^k \cdot k}\right)$

$k = \log_2 n / 2 \qquad O(n)$