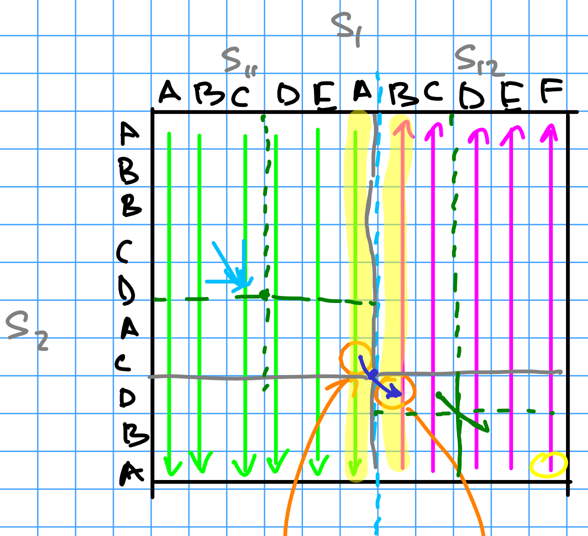
Редакционое расстояние:
Алгоритм Хиршберга

$E(S_1, S_2)$ мы вычисляем за $O(|S_1|\cdot|S_2|)$
за $O(|S_1|\cdot|S_2|)$ памяти

$\exists\ |S_1| = 10^9$
$|S_1|\cdot|S_2| = 10^{18}$

час 3600 с день – 10^5 10^4 дней.



Мы можем за $O(|S_1|+|S_2|)$ памяти посчитать только значение $E(S_1, S_2)$

S_1 = POLYN|OMIAL
S_2 = EXPON|ENTIAL

S_{11} = POLYN
S_{12} = OMIAL

	S_{11}			S_{12}	
Д	–	О	Ж	Д	Ь
Д	Р	О	Ж	–	Ь
	S_{21}			S_{22}	

$E(S_{11}, \text{"ABBCDAC"})$
$E(rev(S_{12}), \text{"ABD"})$

- $\exists\ S_{21}, S_{22} : S_{21} \circ S_{22} = S_2,\quad \underline{E(S_{11}, S_{21})} + \underline{E(S_{12}, S_{22})} = \underline{E(S_1, S_2)}$

- $E(S_1, S_2) = E(rev(S_1), rev(S_2))$

$E(S_{11}, \underbrace{\text{"ABBCDAC"}}_{S_{21}}) + E(S_{12}, \underbrace{\text{"DBA"}}_{S_{22}}) \longrightarrow \min$

$\Rightarrow$ найдём такое ребро за $O(|S_1|\cdot|S_2|)$ времени
за $O(|S_1|+|S_2|)$ памяти

Вызываемся рекурсивно для подзадач.

Время: $|S_1|\cdot|S_2| + \frac{|S_1|\cdot|S_2|}{2} + \frac{|S_1|\cdot|S_2|}{4} + \ldots + = 2|S_1|\cdot|S_2|$

Память: $\mathrm{Max}(|S_1|+|S_2|) = O(|S_1|+|S_2|)$

RMQ и LCA

I. Динамическая постановка RMQ
RANGE MINIMUM QUERY

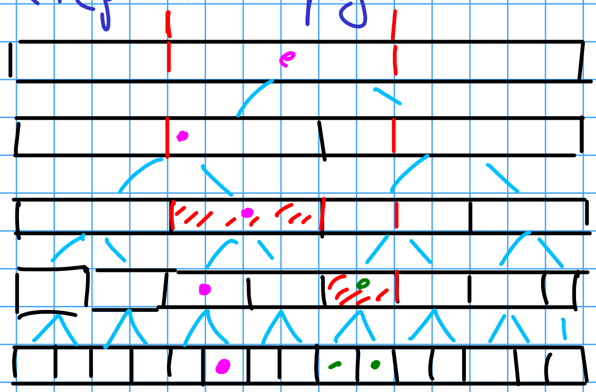
Дан массив данных n

Запрос: 1. $Min(i, j)$ — минимум на отрезке от i до j

2. $Change(i, x)$ — заменить эл-т i на x.

1. $Min(i, j)$ за $O(j-i) \sim O(n)$
   $Change(i, x)$ за $O(1)$

2. Дерево отрезков



$[1, n]$

у отрезка $[k, L]$ сыновья $[k, m]$, $[m+1, L]$

$$m = \left\lfloor \frac{k+L}{2} \right\rfloor$$

Вопрос 1: кол-во узлов в дереве?

$$\sum \frac{n}{2^i} = O(n)$$

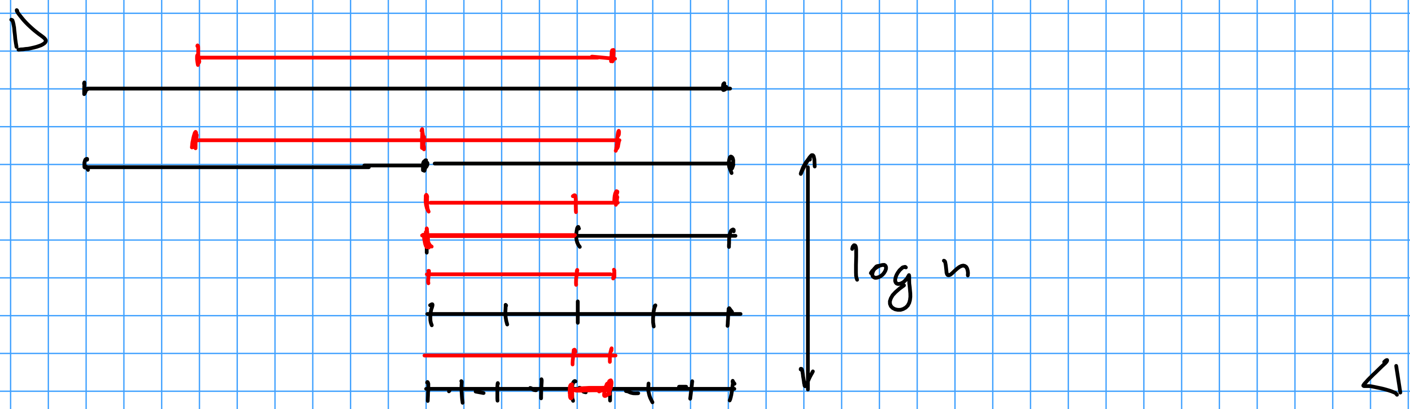
Заполним в $\forall$ отрезке дерева свой min за $O(n)$ (снизу вверх)

Вопрос 2: Как обновлять?

$\forall$ эл-т входит только в $O(\log n)$ отрезков дерева (по одному на $\forall$ уровне)

За $O(\log n)$

Утв: $\forall$ отрезок $= \bigsqcup O(\log n)$ отрезков дерева



- Дерево отрезков работает для $\forall$ ассоц. операции
- Можно реализовать на массиве (как кучу)

II Статическая формулировка RMQ

Массив длины n.

Запрос: $\min(i, j)$

(т.е. массив не меняется)

Давайте отдельно учитывать препроцессинг

$$(f(n), g(n))$$

препроцессинг   запрос

1. Прямое вычисление $(O(1), O(n))$
2. Полное предвычисление $(O(n^2), O(1))$, $O(n^2)$ памяти
3. Дерево отрезков $(O(n), O(\log n))$

RSQ (RANGE SUM QUERY)

Давайте вычислим $S[1:n]$: $S[i] = \sum_{k=1}^{i} a_k$

$$RSQ(i, j) = S[j] - S[i-1]$$

Решение за $(O(n), O(1))$, $O(n)$ памяти

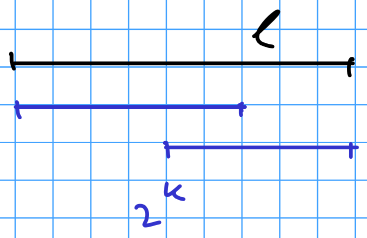
NB: у min нет обратной операции

4. Разреженная таблица.

Давайте вычислим min для всех отрезков длины 2^k.

Утв: $\forall$ отрезок можно представить как объединение двух отрезков длины 2^k.

$\forall\, l \ \exists\, k : 2^k \le l < 2^{k+1}$



Отрезков длины 2^k — $O(n)$

Всего отрезков $O(n\log n)$

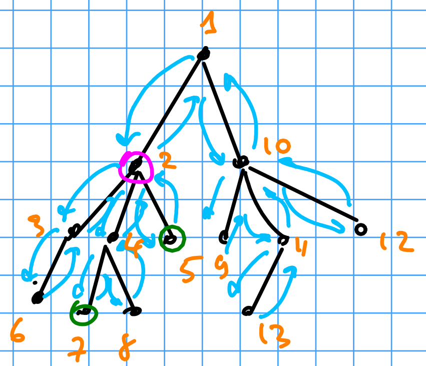
Решение за $(O(n\log n),\ O(1))$

NB: Для идемпотентных операций

$$a \circ a = a$$

NB: таблица для $l \to k$ (LZCNT в x86)

Задача LCA (least common ancestor)



Запрос: (u, v) — пара вершин. Нужно найти ближайшего общего предка.

- Сведём LCA к решению RMQ.

Выпишем порядок вершин в обходе

1 2 3 6 3 2 4 7 4 8 4 2 5 2 1 10 9 10 11 13...

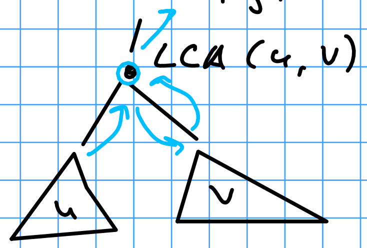
минимальная глубина

LCA (u, v) – ∡ вершины в порядке эйлерова обхода.

найдём первое вхождение u — i

и первое вхождение v — j

Утв: RMQ(i, j) = LCA(u, v)



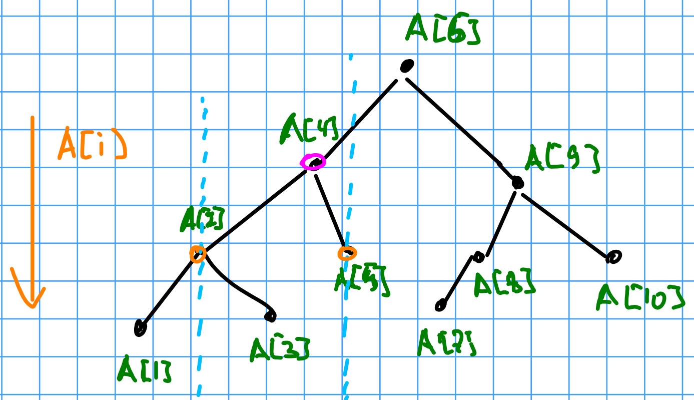
⇒ Все те же алгоритмы для LCA

Например, (O(n log n), O(1)) методом разреж. табл.

• Сведение RMQ к LCA

A[1:n] → (1, A[1]), (2, A[2]), (3, A[3]) ...

Построим декартово дерево



RMQ(2, 5) = LCA(2, 5)

Мы можем: RMQ → LCA → RMQ ...

LCA → RMQ → LCA ...