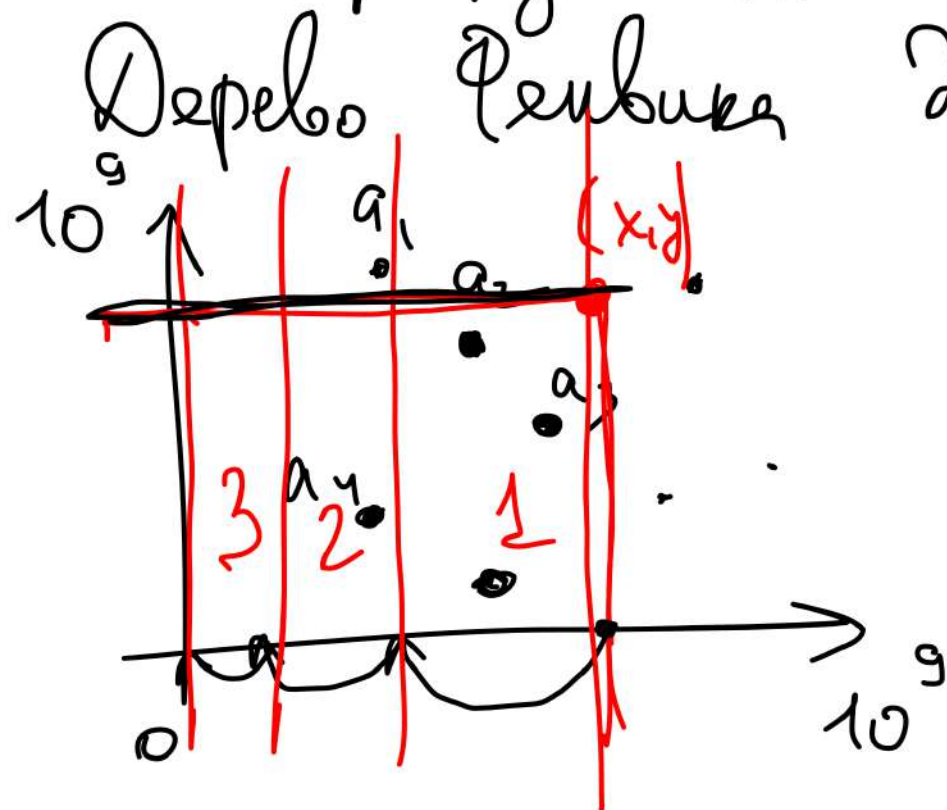


Лекция 11.

18.11



Дерево Ренбука

и тогда

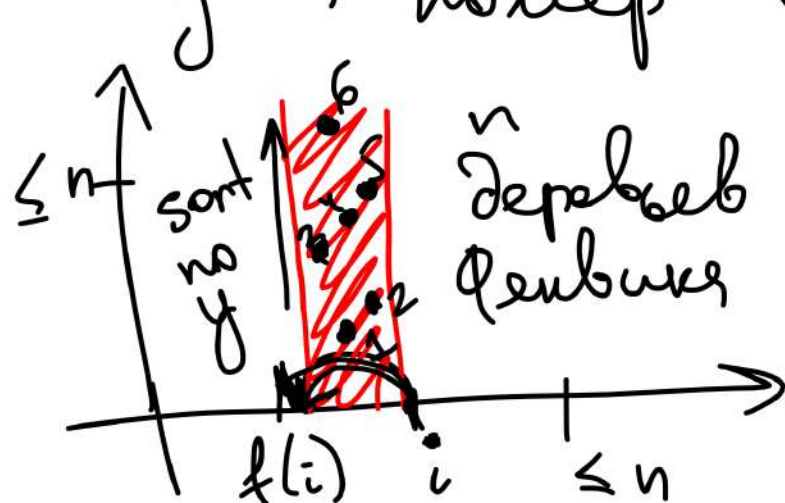
→ узлы a_i

→ сумма a_i в
узлах?

и

Сокмем координаты
 n точек \rightarrow sorted-x: n чисел
 sorted-y: n чисел

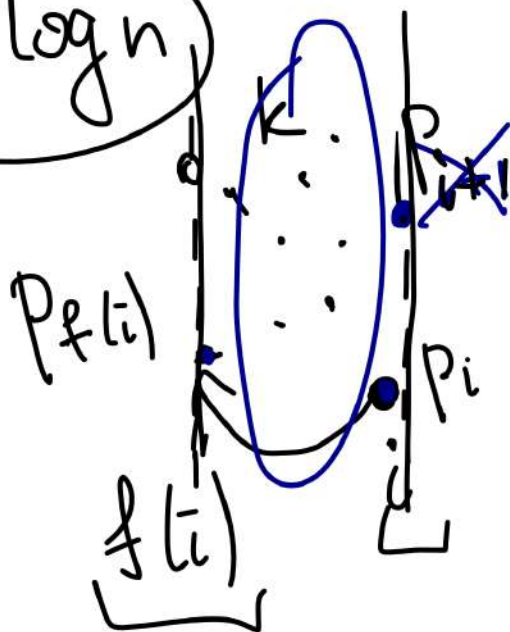
$x \rightarrow$ номер в sorted-x $O(n \log n)$
 $y \rightarrow$ номер в sorted-y $O(n \log n)$



T.e. где каждого i забегает
 отдельную структуру, которая
 хранит все точки в номере
 $\text{num}_x \in [f(i), i]$

Построение Перенос Перенос

$n \log n$



$$k = i - f(i) + 1$$

сортировка

$n \log k$

$p_0, p_1, p_2 \dots p_{n-1}$
 $x_0 \leq x_1 \leq x_2 \leq \dots \leq x_{n-1}$



тогда внутри носим по y
 носим. Перенос: $k \log k$ умножить

Время построения: $n \log n + \sum_i k_i \log k_i$
 $f(i) \leq num \leq i$
 $k_i = i - f(i) + 1$

$$\sum_{i=0}^n k_i \log k_i \leq \left(\sum_{i=0}^{n-1} k_i \right) \log n \leq n \log n$$

$\sum_{i=0}^{n-1} k_i = \sum_{j=0}^{n-1} c_j$, где c_j — кол-во узлов, в которых находится j -я точка

$$c_j \leq \log n$$

j -я точка лежит в i -м окне $\Leftrightarrow f(i) \leq j \leq i$

$$i = j, g(j), g(g(j)), \dots$$

Вывод: построение работает за $O(n \log^2 n)$ и $O(n \log n)$ памяти

Ответ на вопрос

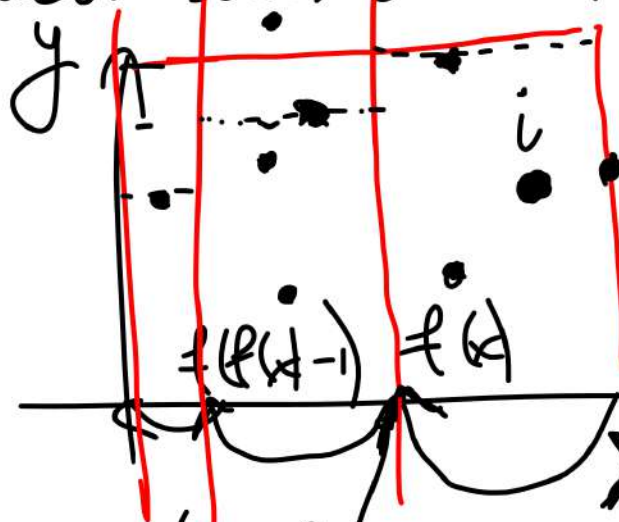
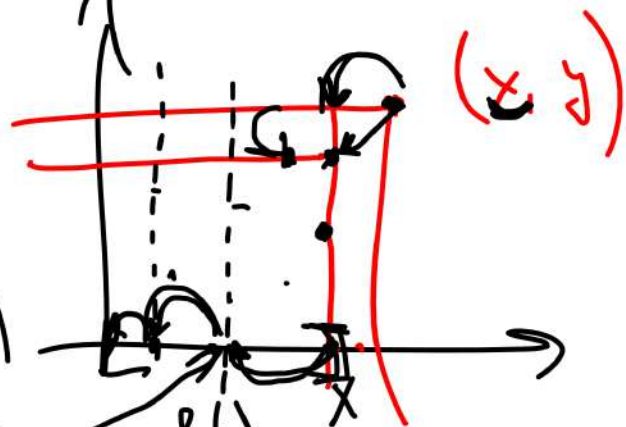
$$k_i = f(l_i) - i + 1$$

(x, y)

$x \rightarrow$

нам в sorted-х

(наиб. число $\leq x$) - бинарный поиск



а именно на пересечении

$\log n$ полос, в каждой полосе бинарный поиск + запрос к Фенвику

$O(\log^2 n)$ на запрос

Дерево поиска.

Храним мн-во S (чисел, например)

→ insert x : добавляю x в S

→ erase x : удаляю x из S

→ find x : сообщить, входит ли x в S .

без повторяющихся м-тов в S

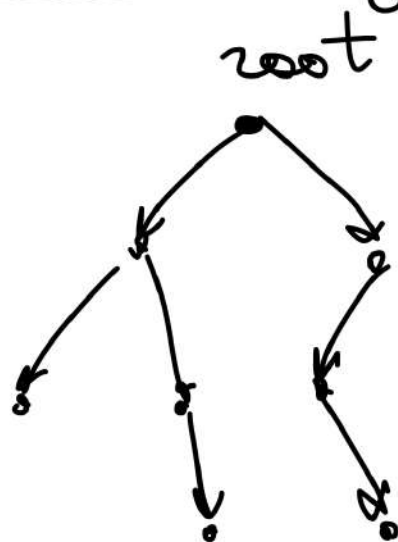
Опционально: → merge S_1, S_2 : объединить S_1 и S_2 в одно новое множество

→ split S x :
 разбить S на
 и на

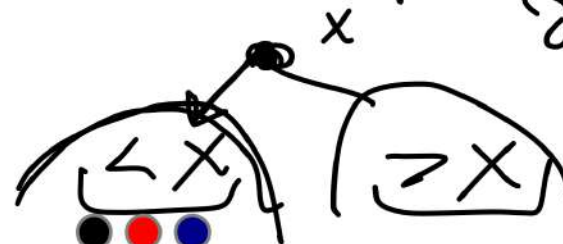
$$S_1 = \{y \in S : y \leq x\}$$

$$S_2 = \{y \in S : y > x\}$$

Двоичное дерево поиска



У каждой вершины не более двух соседей - левый и правый.



Дерево поиска



Наборное дерево поиска.

find x

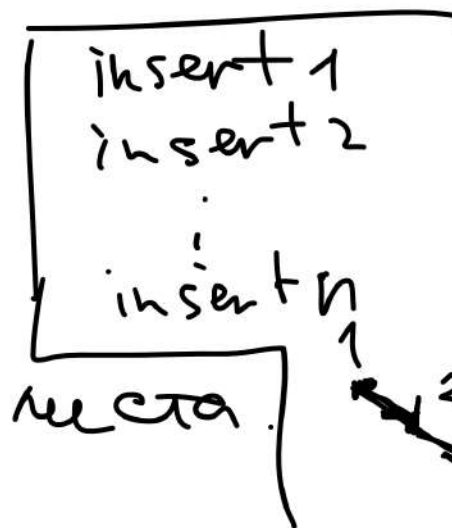
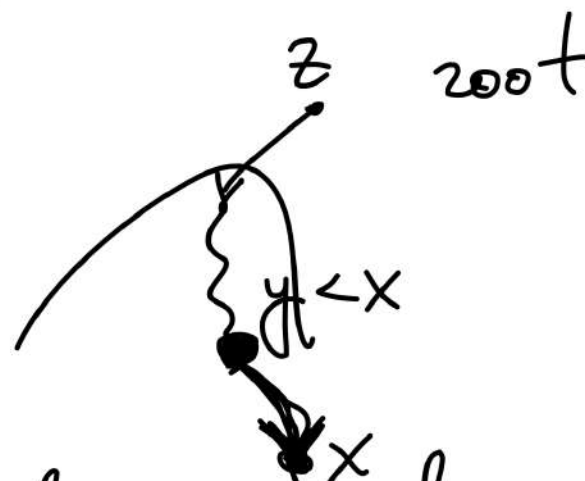
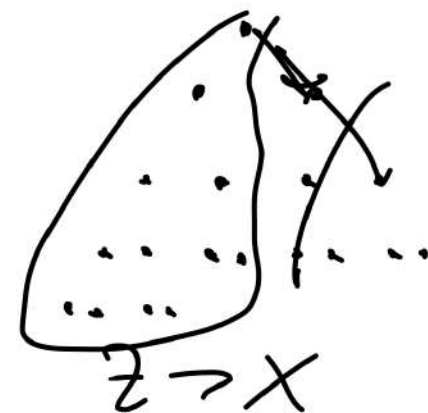
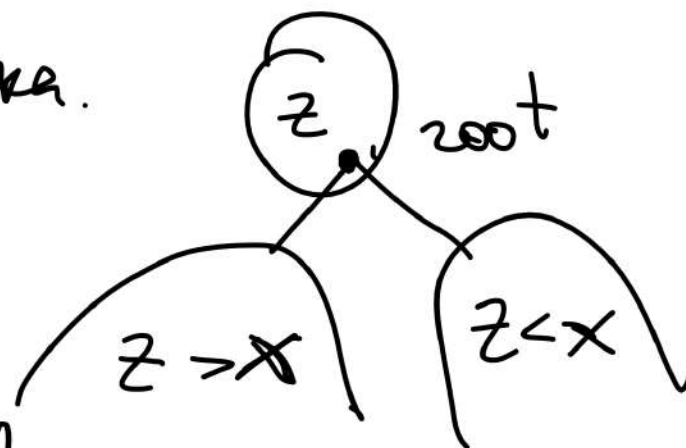
Работает за $O(h)$,
где h — высота дерева

insert x :

find x : если
 x нет в дереве,

то добавляем его.

Указе x добавляется в конкретное место.



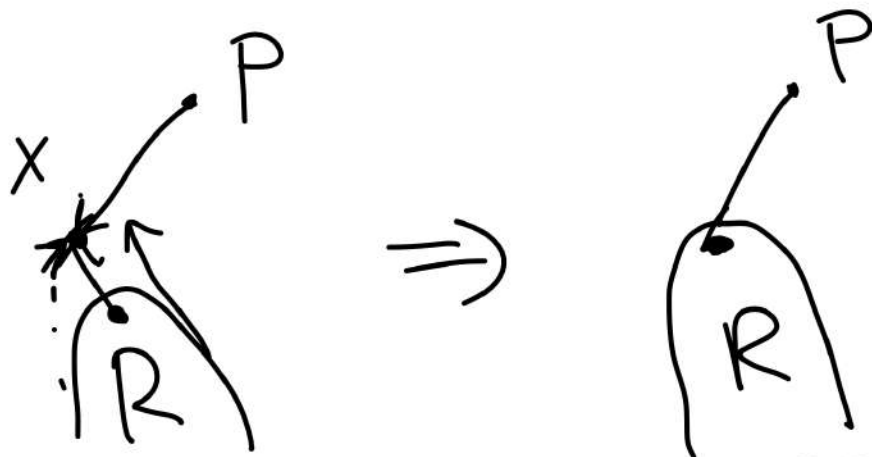
erase x : find x

Если x нет в дереве, то просто вернуть.

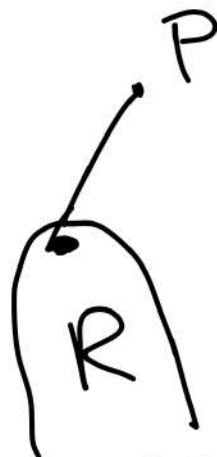


Если x - лист, то просто его отрезаем.

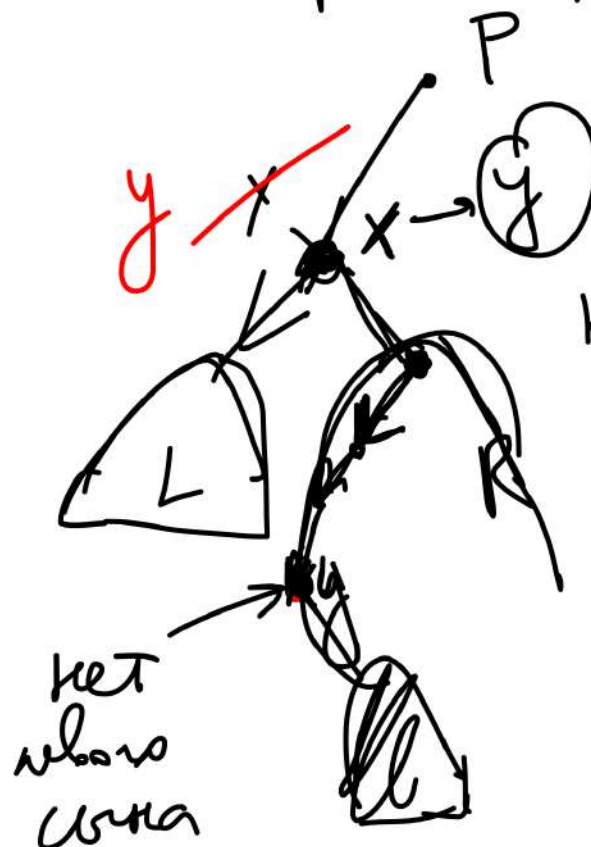
Если $y \neq x$ нет левого сына!



\Rightarrow



Теперь нужно у x либо то $asko$
 либо com (нет $у$ правого)



находим y — самое
 маленькое число в R

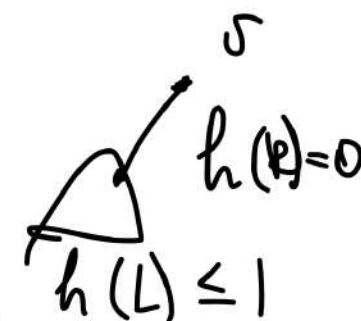
y переписываем
 вместо x ,
 удаляем старое
 вхождение y .



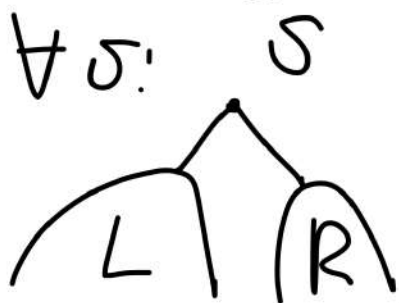
Сбалансированное дерево поиска:
хотим, чтобы глубина была $O(\log n)$,
где $n = |S|$.

AVL-дерево

Адольф-Вельский
Лангс



Дир Если $h(s)$ — глубина
то требуем:



поддерева верш. s ,
 $|h(L) - h(R)| \leq 1$

\Rightarrow AVL-дерево

Какое минимальное количество узлов в AVL-дереве с n вершинами?

Ответ: $S(h)$ - минимальное количество вершин в AVL-дереве высотой h .

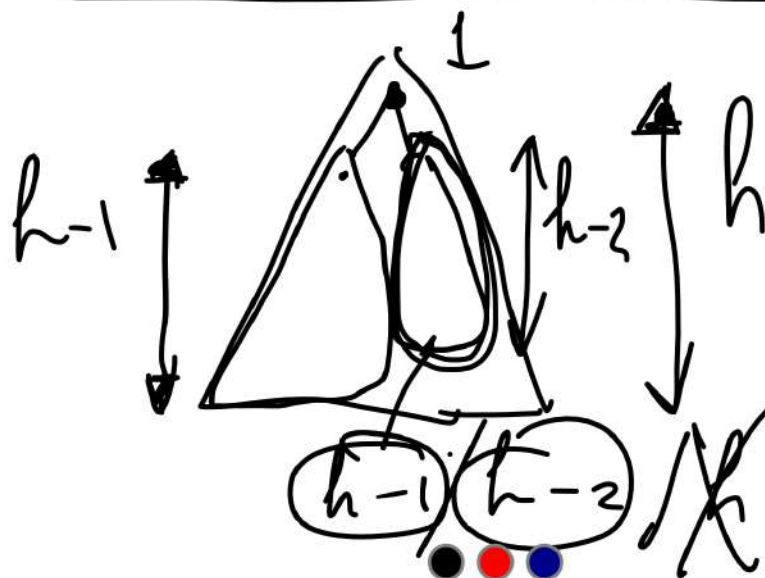
$$S(0) = 0$$

$$S(1) = 1$$

$$S(2) = 2$$



$$S(h) = 1 + S(h-1) + S(h-2)$$



$$S(h) = 1 + S(h-1) + S(h-2)$$

Тога $S(h) = F(h+1) - 1$, яг
 F — мана Фибоначчи, $F(0) = 1, F(1) = 1, \dots, F(2) = 2$
 $F(k) = F(k-1) + F(k-2)$

Ab Ungenauheit von h.

~~base~~ $h=0$ / $h=1$ - over. Dances, next to $h \geq 2$

$$S(h) = 1 + S(h-1) + S(h-2) =$$
$$= 1 + (F(h) - 1) + (F(h-1) - 1) = F(h) + F(h-1) - 1$$
$$F(h+1)$$

$$S(h) = F(h+1) - 1$$

гтв (5/2) Пуско $\varphi = \frac{1+\sqrt{5}}{2}$ $\approx 1,6$

$$F(n-1) = \frac{\varphi^n - (-\varphi)^{-n}}{\sqrt{5}}$$

← берга
генер

$$F(n), F(n+1) \rightarrow F(n+2)$$

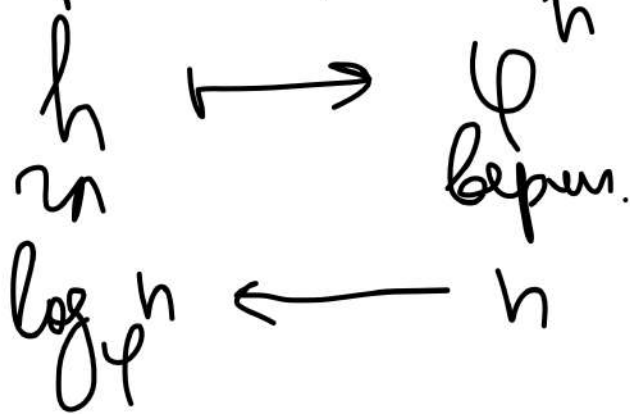
Поэтому $F(n) = \Theta(\varphi^n)$. $\varphi > 1$

$F(n)$ растёт экспоненциально быстро
 $\Rightarrow S(h)$ тоже растёт эксн. быстро, $S(h) = \Theta(\varphi^h)$

Отсюда: если в AVL-дереве всего n вершин, то его высота h удовлетворяет:

$$h \leq O(\log_\phi n) = O(\log n)$$

и

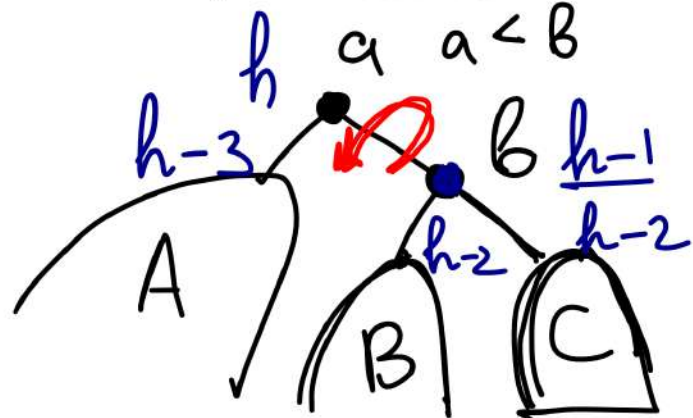
$$\log_\phi n \leftarrow h$$


Def: $\Delta(\sigma) = h(L) - h(R)$



Хорошо, когда $\Delta(\sigma) \in \{-1, 0, 1\}$

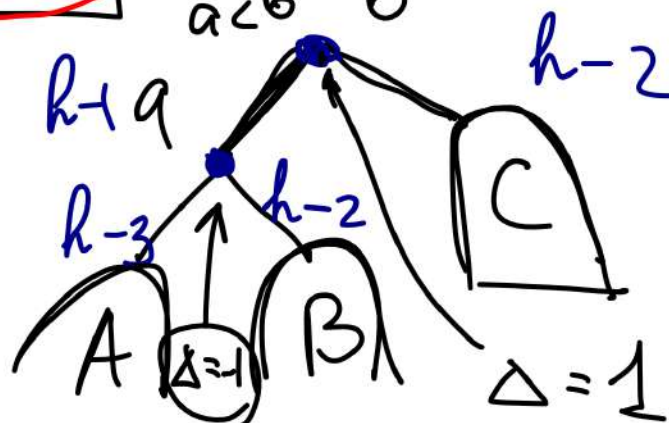
a: $\Delta(a) = -2$



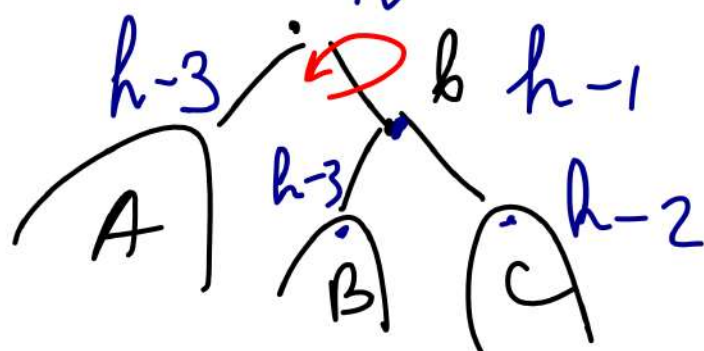
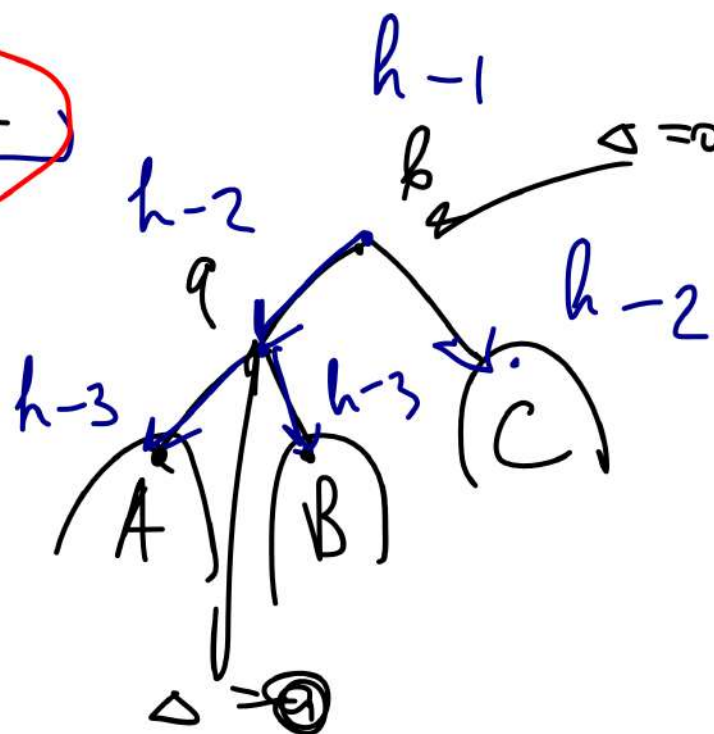
Правильно сориентировать ребро на 2

$\Delta(b) = 0$

сориентировать
ребро на 2
→
дефект
исправить

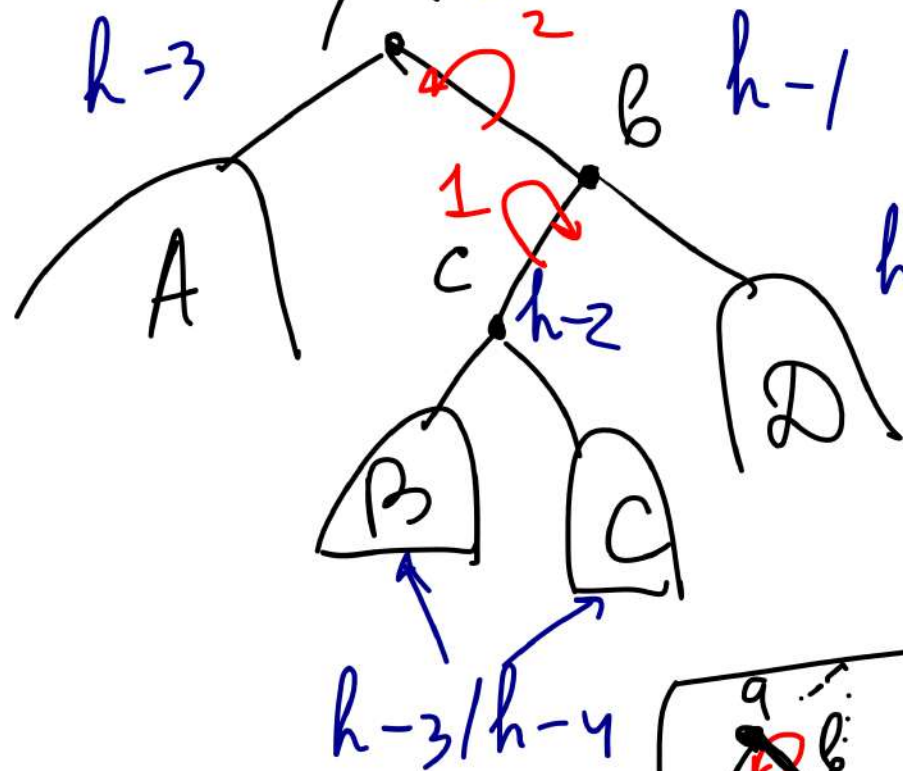


$$\Delta(a) = -2, \quad \Delta(b) = -1$$


 \Rightarrow


$$\Delta(a) = -2, \quad \Delta(b) = 1$$

$\Delta(a) = -2$, $\Delta(b) = 1$, $\Delta(a) = 2$



unbal.
 left
 \Rightarrow
 right
 comp.

