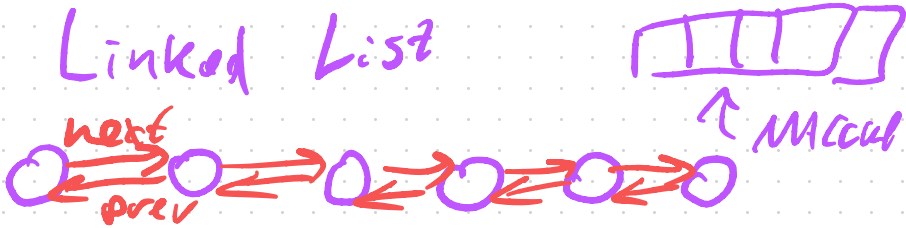
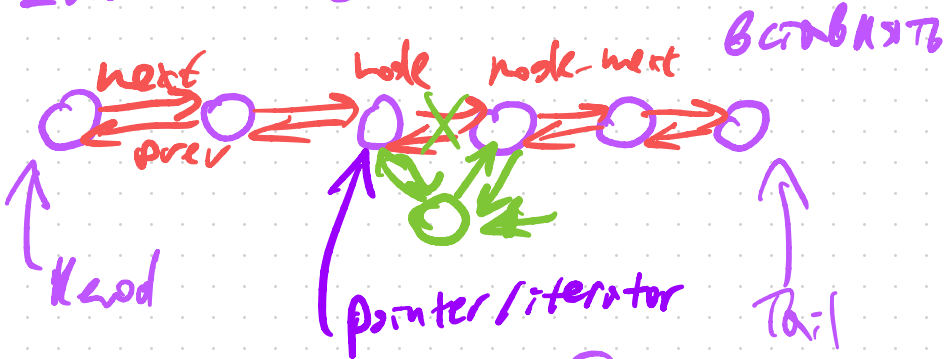


Бинарный поиск и структуры данных



Insert : $O(1)$, если знать куда вставить



List {
Head
Tail

Erase : $O(1)$



```
class Node:
    def __init__(self, node_l, node_r, value=None):
        self.value = value
        self.node_l = node_l
        self.node_r = node_r

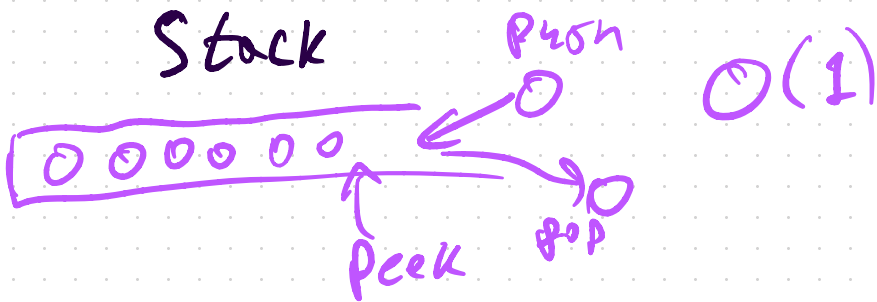
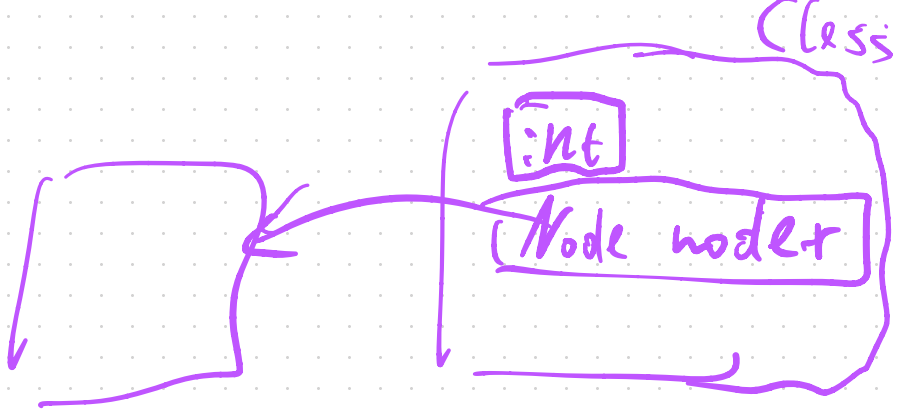
def insert_after(node, value):
    node_next = node.node_r
    node.node_r = node_next.node_l = Node(node, node_next, value)
```



NULL-
Беремка
(Нысгәтә)

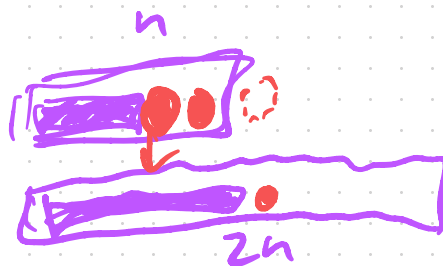
Обход списка

```
→
for (v = head.node_r)
    v ≠ Tail;
    v = v.node_r)
    print(v.value)
```



Варианты: как реализовать

- ① Linked List $O(1)$ в среднем
- ② Vector
(List : [1, 2, 3])
 $O(1)$ амортиз.



push-back

$O(1)$ в среднем
↳ $\begin{cases} 1 \\ n \end{cases}$

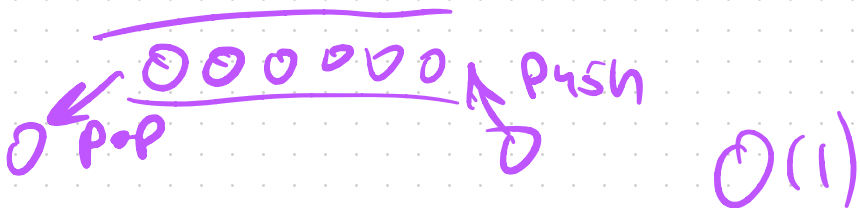
Vector (List Container)

обращение по i : $O(1)$
push-back: $O(1)$ в ср
pop-back: $O(1)$

Linked List

обращение по i : $O(i)$
push-back
pop-back: $O(1)$
insert / erase по
указателю
на beginning $O(1)$

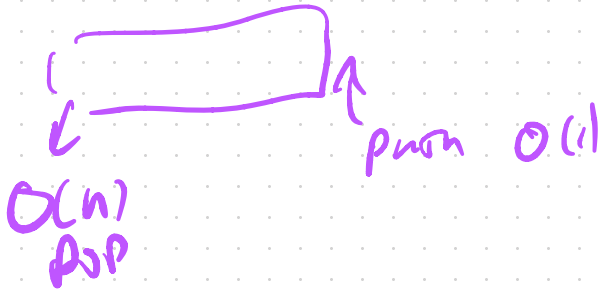
Queue:



Решение:

① Linked List $O(L)$

② Vector



②' Vector ω -yuzue:



push(x)

$arr[r] = x$

$r = (r + 1) \% n$

pop():

result = arr[l]

$l = (l + 1) \% n$

return result



Push u POP: $O(1)$
номер элемента $< N$

реализация



$l = 0$ $[0; 0)$
 $r = 0$

Deque:



реализация как у
одежда

Бинарный поиск:

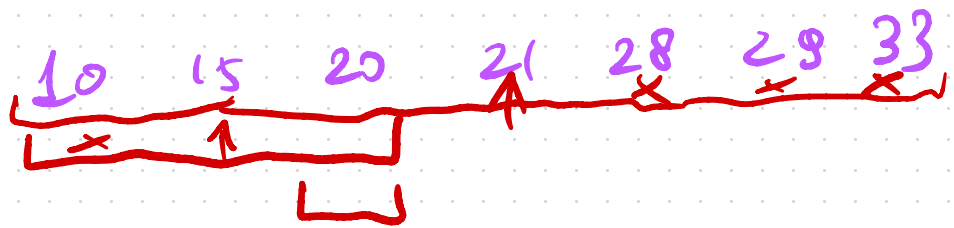
S — множество

Вопрос: $x \in S$?

Как реализовать?

① S — в массиве
Проверка $x \in S$ — $O(n)$

② S : отсортировать



$x = 20$



$$l = -1$$
$$r = n$$

while $l+1 \neq r$:

$$(l, r) =$$
$$= l+1 \dots r-1$$

$a_0 \dots a_{n-1}$

$$m = (l+r)/2$$

if $a_m == x$

return True

if $a_m > x$:

$$r = m$$

$O(\log n)$

else:

$$l = m$$

return false

Бинарный на упорядоченных

lower bound

a_i : 10 15 22 34 39

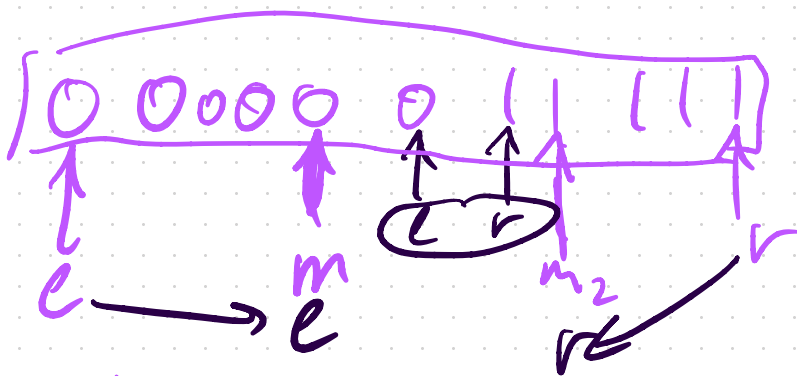
$x = 25$: $\min i: a_i \geq x$

$a_i \geq x$

0 0 0

1 1

↑
нельзя
есть



$f(l)$ всегда 0

$f(r)$ всегда 1

Аккуратно с крайними
случаями!!!

$a = \quad 10 \quad 20 \quad 30$

$x = 5$

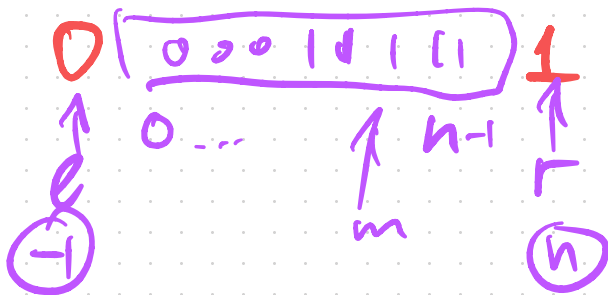
$a_i \geq x$



\Rightarrow случается ошибка



\in группы крайних
случаев



$l = -1$

$r = h$

while $r - l > 1$:

$m = (l + r) // 2$

if $(a[m] > x)$:

$r = m$

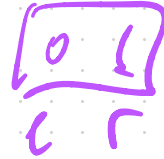
else

$l = m$

return r

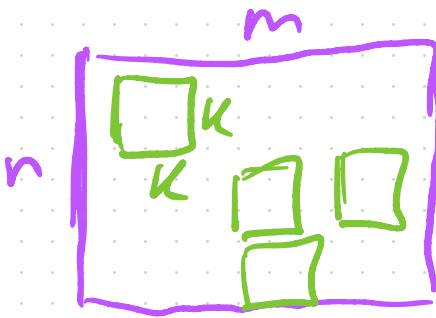
были значения
и результаты

$f(m)$



Ближе к

"но отбери"



$l < m$

МАКСИМАЛЬНЫЕ

k ,

это

поместиться

N карт

$k = 5 \text{ cm}$ - то же OK



$k = 4 \text{ cm}$ - тоже не подходит

11110 00000
 ↑
 k

$L = 0$ - то же не подходит

$R = \infty$ ($n+1$) - то же не подходит

while $R - L > 1$:

$mid = (L + R) // 2$

 if $\lfloor \frac{n}{mid} \rfloor \cdot \lfloor \frac{m}{mid} \rfloor \geq N$:

$L = mid$

 else:

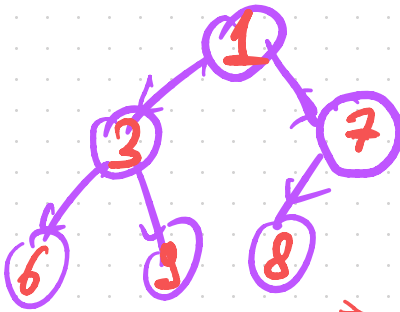
$R = mid$

return L

Binary Heap

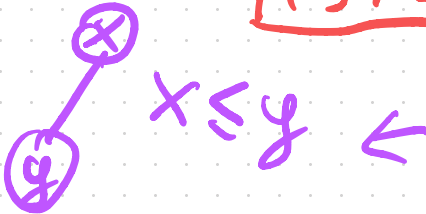
1964

W Williams

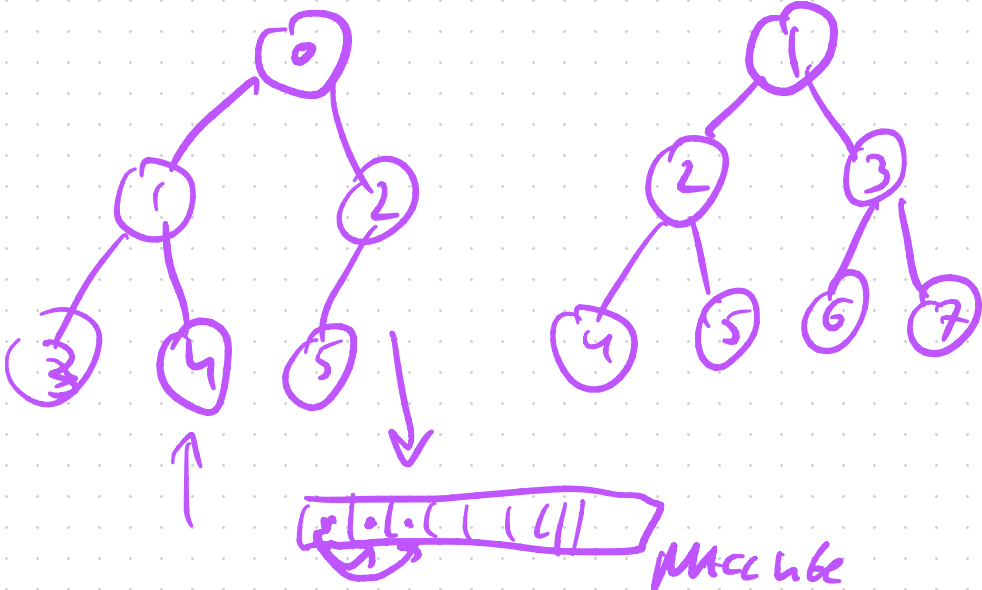


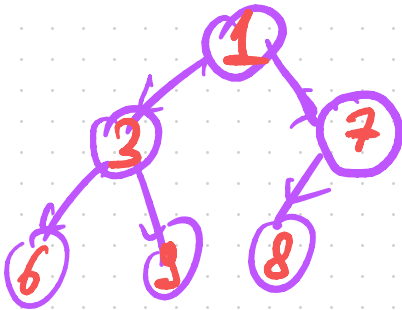
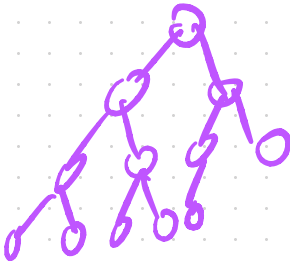
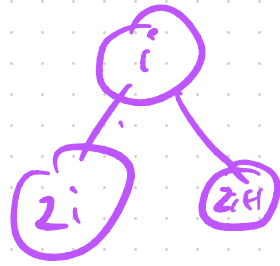
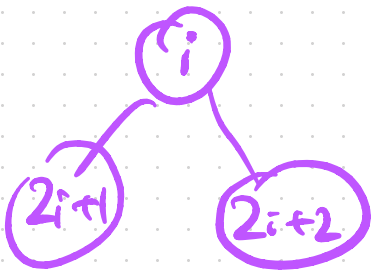
- почти
полное
бинарное
дерево

[1 3 7 6 5 8]



Как хранить?





Min(): $O(1)$
return $a[0]$

MAX()

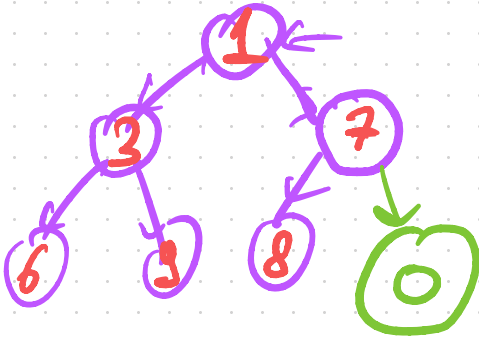
Q^k
 Q^y Q^z

can Kyu

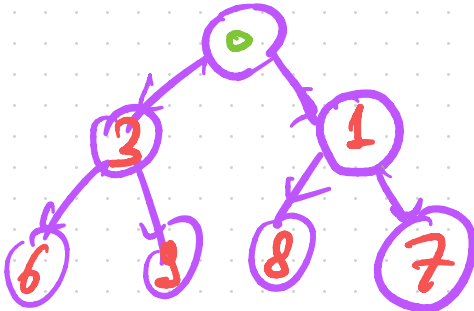
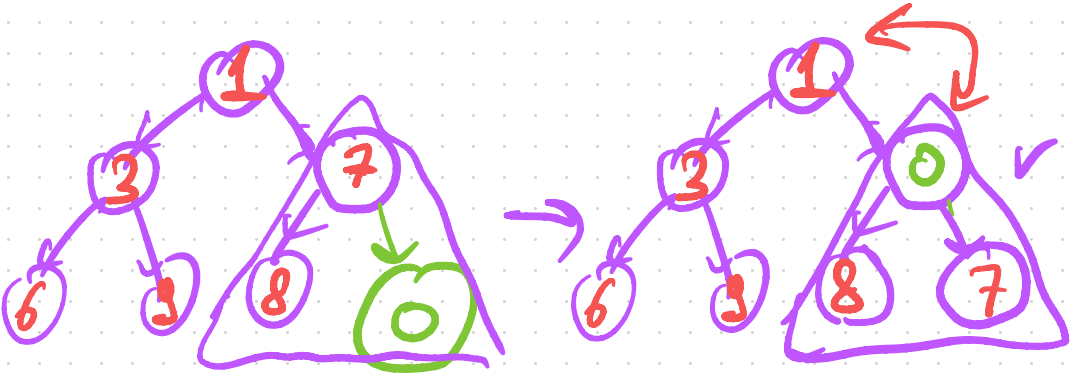
na MAX

Добавление элемента.

1) в. push-баса(x)



2) показать кабарухит кузи



Sift Up (i)

while $i \neq 0$ and

$O(\log n)$ $a[\lfloor \frac{i-1}{2} \rfloor] > a[i]$:

SWAP($a[i]$, $a[\lfloor \frac{i-1}{2} \rfloor]$)

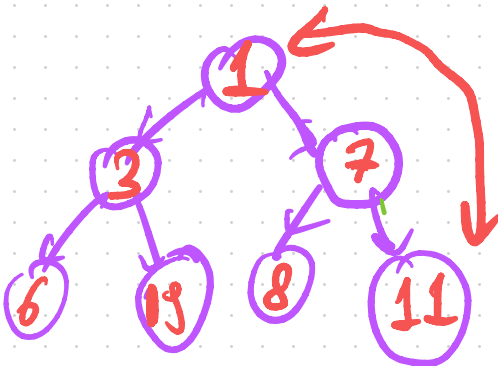
$i = \lfloor \frac{i-1}{2} \rfloor$

Add(x):

a. push-back(x)

siftUp($\text{len}(a)-1$)

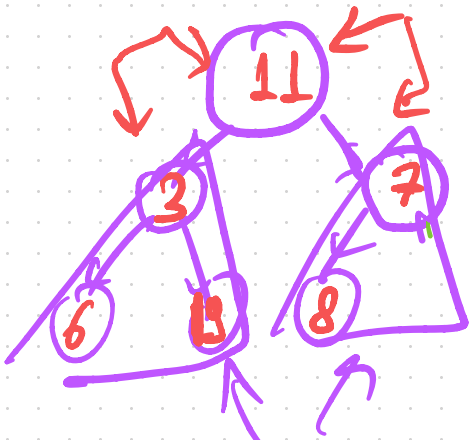
Extract Min():



$n = \text{len}(a)$

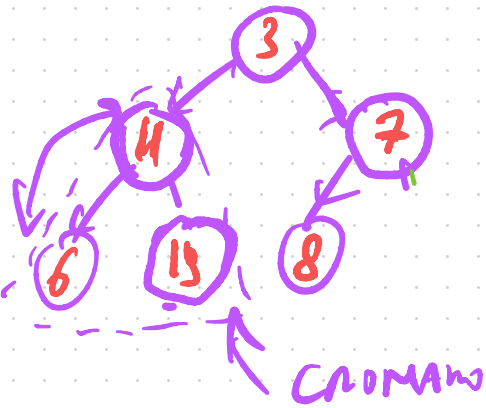
SWAP($a[0]$, $a[n-1]$)

a. pop-back()

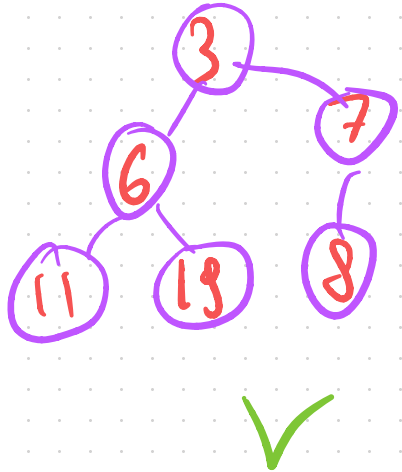


Корректная куча

⇒ min у нас 3, у нас 7



⇒



Не все эти
элементы могут

Sift Down (i)

while True:

$j = \operatorname{argmin} \{ a_i$

a_{2i+1}

$a_{2i+2} \}$

$O(\log n)$

if $j == i$:

break

else

SWAP(a_i, a_j)

$i = j$

ExtractMin() $n = \text{len}(a)$

SWAP($a[0], a[n-1]$)

$a.\text{pop-back}()$

SiftDown(0)

Построение кучи:

Метод 1:

$(3, 10, 15)$ → куча?

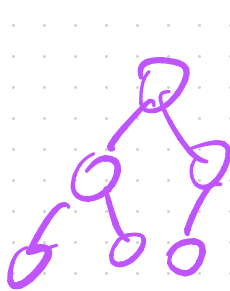
for (n)
odd(i)

$\Theta(n \log n)$

push-back 1
+ siftUP log

Метод 2:

0.. n-1

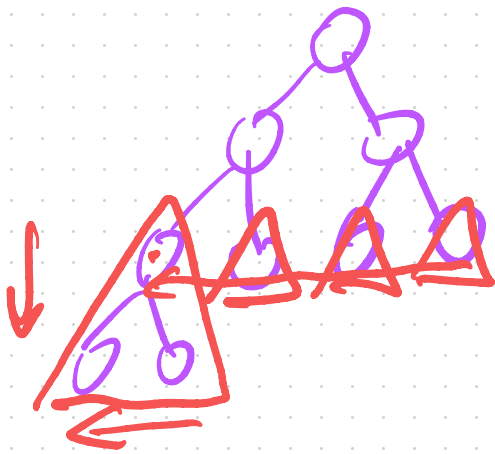


1) считываю
это массив
это куча

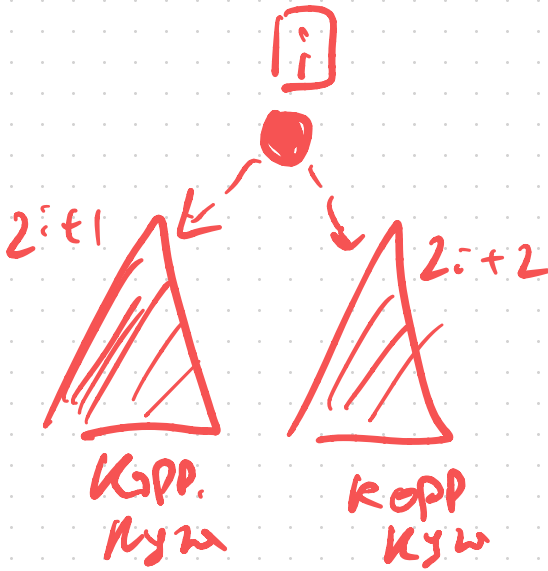
2) начинаю
сбавивать

for i = n-1 .. 0: ← log
siftDown(i)

$\Theta(n)$



УТВ: В поле бинаре
 если мы
 прошли $[n-1, x]$,
 то каждая эл-т
 в $[n-1, x]$
 образует кдрр.
 куду со своим
 поддеревом



Сортировка кудей

$a[0 \dots n-1] \leftarrow$

$\rightarrow \text{BuildHeap}(a) \quad O(n)$

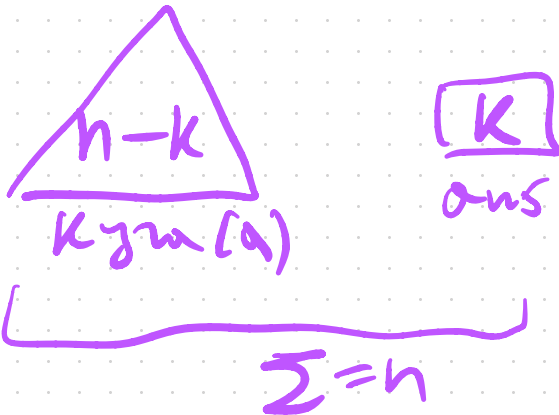
$\rightarrow \text{ans} = []$

$O(n \log n)$ $\left(\begin{array}{l} \text{for } (i = 0 \dots n-1) \\ \text{ans.append}(\text{extract Min}(a)) \\ \end{array} \right)$

Time: $O(k \log n)$

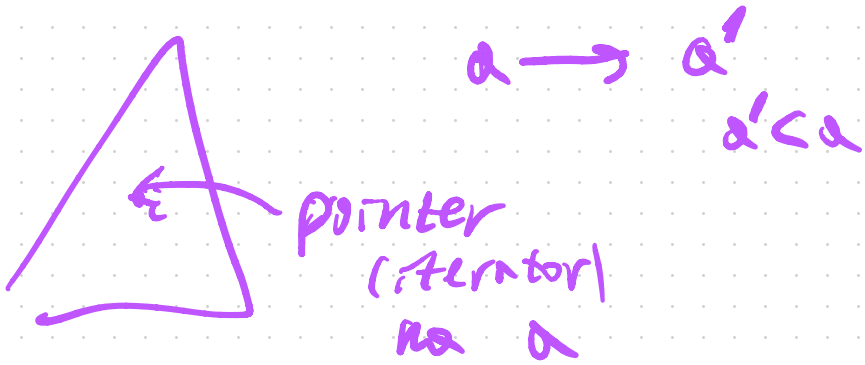
Memory: $O(k)$ ans
loop $O(n)$

max ko chupno $\therefore O(1)$ karta



Stability: \rightarrow

Decrease Key



decreasekey (given n , a , a')

$k[a] = a \rightarrow a'$ ||
Sift up
||

$O(\log n)$