

| Today's ag | enda | | |
|------------|------------------------|-------------|-----------|
| 4 LI | enda utorduction to | Heap/Pa. | |
| | smallest ele | | |
| | | aray Xleete | ode hold] |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| ΔΙα | o P | ror | |
| AI 9 | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |



11 Introduction

| | inlev (n) | getmin () | delete min() |
|----------------|-----------------------------|--|--|
| Averaglist | 0(1) | 0(N) | (4)0 |
| Linkedlis4 | olas) | 0(N) | 0(N) |
| quene | 0(1) | 0(m) | 0(~) |
| Hashmap | 0(1) | 060) | 0(4) |
| | no. of ell | pa o(1) | 1 |
| Pa | O(logn) | 0(1) | Ologi |
| PQ | O(logn) | Priority Queve < Inter | er) Pa= nev |
| PQ Min Pl | | | |
| Priority Queve | < Integer> PQ = neu | Priority Queve < Intermediate | per> PR= new Priority Queue < Collections. |
| Priority Queve | Integer> Pa: neu Poionit | Priority Queve < Integral Man R2 | per> PR= new Priority Queue < Collections. |
| Priority Queve | < Integer> Pa: new Priority | Priority Queve < Integrand Man Pa | per> PQ = New PriorityQueue Collections. |
| Priority Queve | Integer> Pa: neu Prionit | Priority Queve < Integral Man Pa | (logn) |
| Priority Queve | Integer> PR= ner Poionit | Priority Queve < Integral Man PR y Queue < >(); | (logn) (logn) |



| 0) Kth Smallest Clement |
|--|
| Griven a distinct elements, Point & Smallest elements. |
| En; arr[10]: 183104112761413 |
| anos[3] = {-3 6 2 0 8 7 10 4} |
| K:3: -3 0 2 |
| Videa! Us Sort the away and return the first K elements. T.c.: O(vlog N) |
| 11 idea2 Is Add all elements to min PQ and get the first k elements. |
| T.c:o(NlogN) |
| |
| |
| |
| |

> K Smaller elements



| Nidea3 | |
|--|--|
| apolio]: { 8 3 1 | 10 4 11 2 7 6 14 13 |
| K:3 | |
| 4123 | eman heaf |
| | 2 |
| | |
| | 3 1 |
| | |
| 119Suedo code | |
| months and | Constlest dhemalh |
| 2 | Smallest dhundh |
| Land William Committee | weakest responses |
| void Kth Smallest (int | arolin, int k) { mh: |
| mantleap < int > 1 | |
| | |
| 11 /2 2 2 | |
| for lint iso; i | < x; i++) \(\tau \cdot |
| for lint iso; inheadd | (avolis); T.c: O(NH logh) (avolis); S.c: O(K) |
| mh.add | (awfil); S.c. O(k) |
| Jor (ind is K; | (amlis); S.c: O(K) |
| Jor (ind is K; | (asoli); S.c.: O(K) i <n; <="" asoli)="" i++){="" mh.="" peck())="" th="" {<=""></n;> |
| Jor (ind is K; | (awfil); S.c.: O(K) i <n; <="" awfil="" i++){="" mh.="" peck()){="" remove();<="" th=""></n;> |
| Jor (ind is K; | (asoli); S.c.: O(K) i <n; <="" asoli)="" i++){="" mh.="" peck())="" th="" {<=""></n;> |
| Jos (ind is Kg | (awfil); S.c: O(K) i <n; (awfil);<="" <="" add="" awfil="" beek()){="" i+1){="" mh.="" remove();="" th=""></n;> |
| Jor (ind is K; | (avolis); S.C: O(K) i <n; (="" (avolis);="" <="" add="" avolis="" contince;)<="" i++){="" mh.="" peek()){="" se="" th="" vemove();=""></n;> |
| Jos (ind is Ks) jos (ind is Ks) jos (ind is Ks) jos (ind is Ks) | (awolis); i< M; i++) \ avolis < mh. peck()) \ mh. add (avolis); Mh. Size() >0) \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() |
| Jor (ind is Kg | (awolis); i< M; i++) \ avolis < mh. peck()) \ mh. add (avolis); Mh. Size() >0) \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() \ mh. Size() >0 \ mh. Size() >0 \ mh. Size() |
| Jor (ind is Kg | (avolis); i< N; i++) \ avolis < mh. peek()) \ mh. vemove(); mh. add (avolis); se \ (continue;) |



| | | J |
|-----|----|-----|
| | 10 | 14 |
| 46 | oP | rep |
| 719 | | |
| | | |
| | | |



11 median

La middle element of sorted array.

a = (3): {2 5 3} 6 (2,3,5) → 3

ams]: {436853 44345683→5

000 (6): {4395122} 4 {23459123345.45

a = 54]: $\begin{cases} 4 & 6 & 10 & 14 \end{cases}$ $4 & 6 & 10 & 14 \end{cases}$ $4 & 6 + 10 = \frac{16}{2} = 8$

Break till 9:28 Pm

rect Code 235



| Q) | Point | median | after | each | insection. |
|-----------|-------|--------|-------|------|------------|
| | | | I/ | | |

arks: 9 6 3 10 4 9 7.5 6 7.5 6

llideas

Ly After each insertion, sout the array & return the middle one.

T.c: N= NlogH 2 O (n2logn)

Midea 2

3 5 11 23 20 25 17 7

3 7 man min 17 20
11 5 23 25

* if total elements are even?

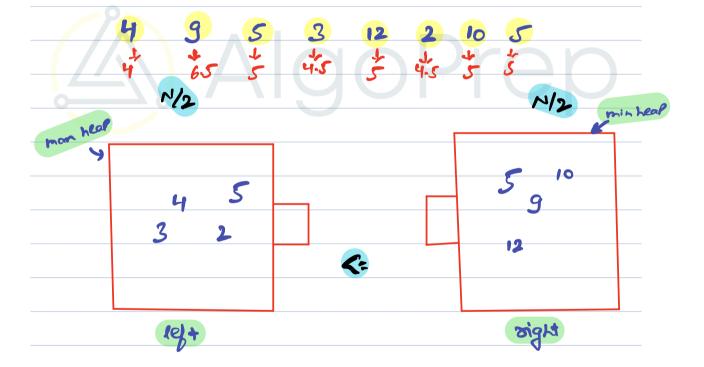
Grandian: man of left bucket + min of right



3 5 11 23 20 25 17



if total elements are odd: Is median = man of left bucket





if (left. size c) == right. size c) added to left PQ but to maintain inequality Pass it via sight PQ. il (left. size () 1= hight. size()) 6 ultimately new element should be added to hight PQ but to maintain inequality Pass via 19+ PQ. 11 P. Suedo Cade Class median Finder man Heal < Integer > left 3 minteal < Integer > oight; Public median Finder () { Toco O(n*logn) Public void addrum (int num) { if (left-size() == sight-size())4
hight-add (hum); S.c: O(N) left.od (sign-semove ()); else 4 14t. add (num) right-ad (1ext-remove())



| 3 | |
|------|---|
| else | double findmedian () { left-size() == bight-size()) / Jouble and = (left-feck() + bight feek()) 2.0 return and; 24 |
| 3 | OP(ep |
| | |