

## Heap / Priority Queue

23 September 2023 13:05

- Min Heap
- Max Heap

Bank



older  
(max heap) =



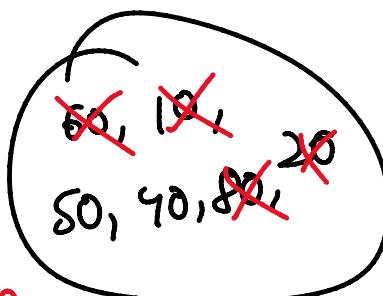
younger  
(min heap) =



~~Max  
Heap~~

~~Min Heap~~

10, 20

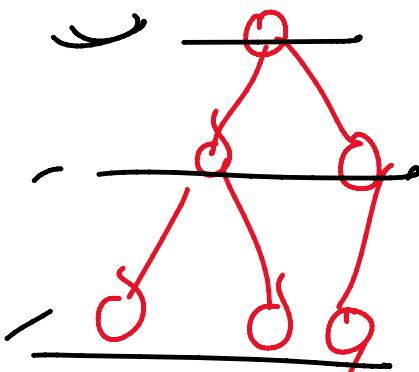
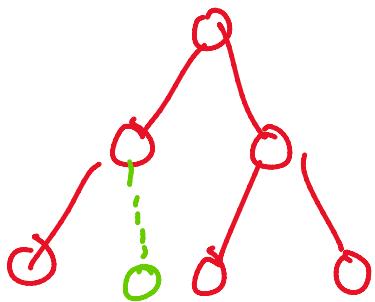


Max Heap  
80, 60,

1. Binary tree (at max 2 children)
2. Complete binary tree.
3. Heap Order property. (Min, Max)

all levels should be completely filled. Except the last  
... and from left to right dir.

all levels should be filled from left to right dir. level. Last level is filled from



insert +  $O(\log N)$   
delete +  $O(\log N)$

top 5 o(1)

C++ (8th)

push

POP

109

Empty

# Java (collection)

add / offer

poll / remove

peek

isEmpty

## C++ ÷

Ch 11. ~~Max Heap~~ priority-queue <int> pq;

Max Heap priority-queue `int, vector<int>, greater<int>`

# Javat

$\tau, \dots, \tau_n = \text{newPriority}(\tau_{n-1}, \chi)$

Java

Min Heap

Max Heap

PriorityQueue<Integer> pq = new PriorityQueue<X>();

(Collection  
interface)

## Kth Largest element in an array

23 September 2023 13:21

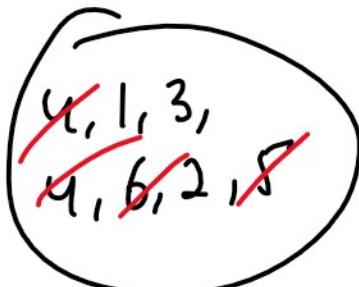
$$\text{arr} = [4, 1, 3, \cancel{4}, \cancel{6}, 2, \cancel{5}], \quad k=4$$

$$\text{ans} = 4$$

- ~~1) Sort the array . And find  $k^{\text{th}}$  value from last~~
- ~~2) Max Heap.  $\rightarrow O(N \log N)$~~
- ~~3) Min Heap  $\rightarrow O(N \log k)$~~
- ~~$\rightarrow O(k)$~~

Time  $\leftarrow O(N \log N)$   
Space  $\leftarrow O(1)$

Max Heapt



Heap size is  $N$

$k=4$

$6, 5, 4, 4 \rightarrow \underline{\text{ans} = 4}$

Time  $\leftarrow O(N \log N)$   
Space  $\leftarrow O(N)$

min Heapt

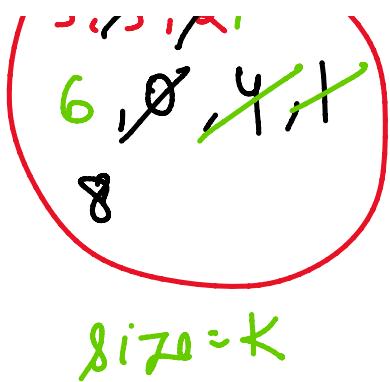
$\text{arr}:$



$k=3$



~~ans = 2 3 4 5~~



~~ans: 2 3 4 5~~

space =  $O(k)$   
time =  $O(N \log k)$

1.  $k^{\text{th}}$  largest  $\rightarrow$  Min Heap  
 2.  $k^{\text{th}}$  smallest  $\rightarrow$  Max Heap

↓  
Closest

## Min cost of ropes

23 September 2023 13:43

~~ropes[] = [4, 3, 2, 6]~~  
~~7   9   15~~

ans = 29

$$\text{cost} = \cancel{\emptyset} + 7 + 9 + 15 = 31 \quad (\text{minimize this})$$

~~ropes[] = [4, 3, 2, 6]~~  
~~7   9   5   15~~

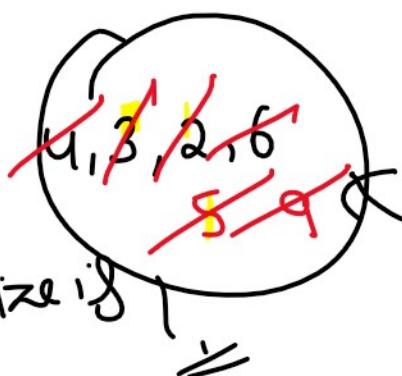
$$\text{cost} = \cancel{\emptyset} 5 + 9 + 15 = 29$$

Two minimum length ropes.

~~[4, 3, 2, 6]~~  
~~5   9~~  
~~15~~

$$\begin{aligned}\text{cost} &= 5 + 9 + 15 \\ &= 29\end{aligned}$$

Min heap!



$$\begin{aligned}\text{cost} &= 5 + 9 + \\ &15 = 29\end{aligned}$$

fill the heap size 3

~~(4, 5, 6, 7, 8)~~

~~cost = q + 1~~ +

proposet: ~~[4, 5, 6, 7, 8]~~

9 + 13 +

cost = 9 + 13 +

Space = O(N)  
Time = O(N log N)

K closest points to origin

23 September 2023 14:06

points [] [] =  $\begin{bmatrix} 3, 3 \end{bmatrix}, \begin{bmatrix} 5, -1 \end{bmatrix}, \begin{bmatrix} -2, 4 \end{bmatrix}$   $k = 2$

$$(3-0)^2 + (3-0)^2 = 3^2 + 3^2 = 18$$

$$\begin{aligned} (5-0)^2 + (-1-0)^2 &= 5^2 + 1^2 = 26 \\ &= (-2)^2 + (4)^2 = 20 \end{aligned}$$

ans =  $\begin{bmatrix} 3, 3 \end{bmatrix}, \begin{bmatrix} -2, 4 \end{bmatrix}$



~~(array + heap)~~

pair of pair

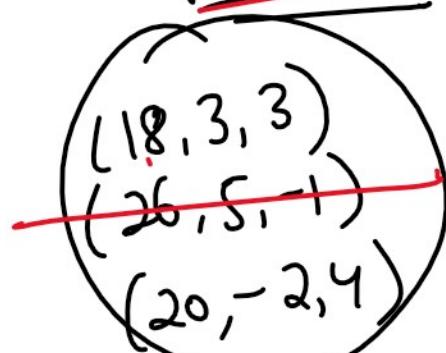
pair<int, pair<int, int>

~~( $x, y$ )~~

distance, x-coor, y-coor,

max heap

ans =  $\begin{bmatrix} 3, 3 \end{bmatrix}, \begin{bmatrix} -2, 4 \end{bmatrix}$



Space =  $O(k)$   
Time =  $O(N \log k)$

Comparator

o. time

$\underline{\underline{[1, 2, 6, 7, 0, 3]}}$

Comparator

Sorting

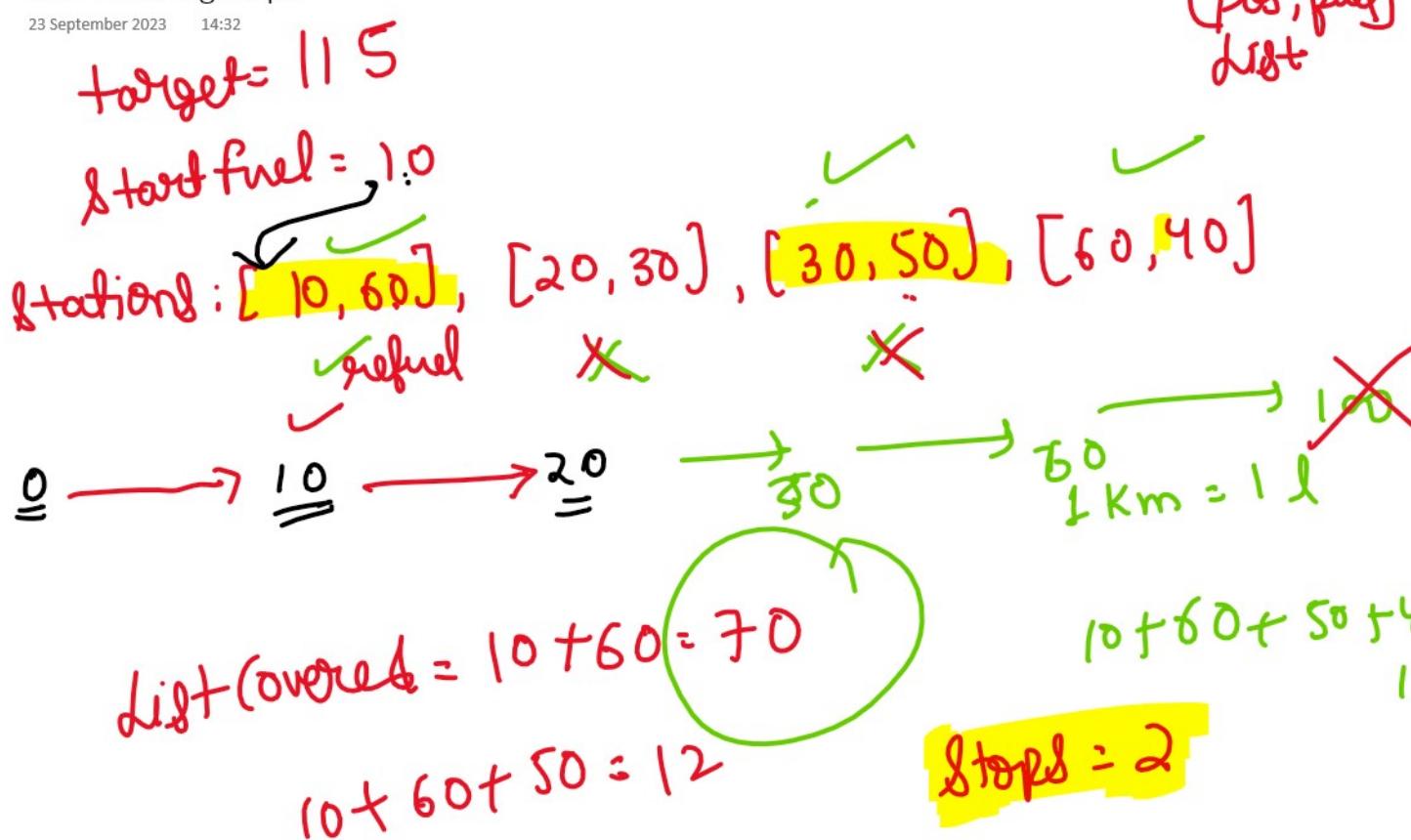
dec. order

0, 1, 2, 3, 6, 7

7, 6, 3, 2, 0, 1

# Min refueling stops

23 September 2023 14:32



$$\text{target} = 200$$

$$\text{start fuel} = 10$$

[5, 30], [10, 20], [40, 70], [60, 60], [100, 90]

$$\text{stops} = 3$$

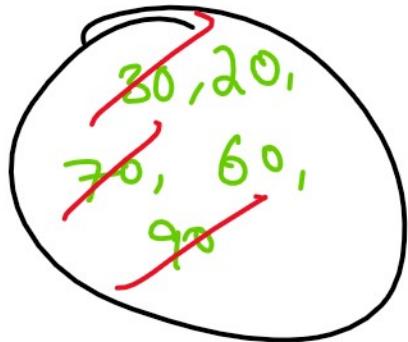
$$\begin{aligned} \text{dist (overall)} &= 10 + 30 + 70 + 90 \\ &= 200 \end{aligned}$$

mid sleep

$$\text{max dist covered} = 10 + 30 + 70 + 90 = 200$$

~~stops = 0 / 1 / 2 / 3~~

$$\text{target} = 200, \quad \text{start fuel} = 10$$



$[5, 30], [10, 20], [40, 70], [60, 60], [100, 90]$  q1

$$\text{max dist covered} = 10 + 30 + 70 + 90 = 200$$

~~stops = 0 / 1 / 2 / 3~~

220



Space  $\rightarrow O(N)$   
Time  $\rightarrow O(N \log N)$

$$\text{target} = 100 \text{ or } 550$$

$$\text{start} = 10$$

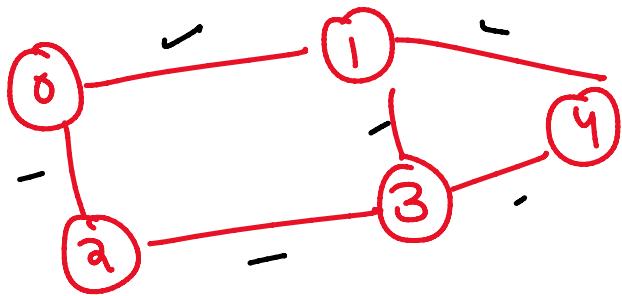
[ ]

$[10, 200], [50, 600]$ , ✓

~~X~~

## Graph intro

23 September 2023 15:26



1. Vertices/nodes
2. Edges.

vertices: 0, 1, 2, 3, 4

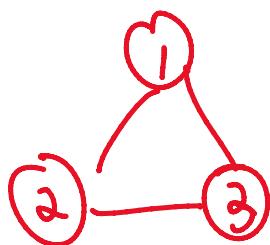
Edges: 0-1, 0-2, 2-3, 3-1, 3-4, 1-4

Real life examples:

① Social Media (Insta, FB)

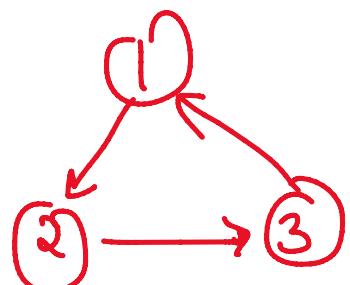
② Transportation

① Undirected / Directed



undirected

1-2 (2-1)  
1-3 (3-1)  
2-3 (3-2)

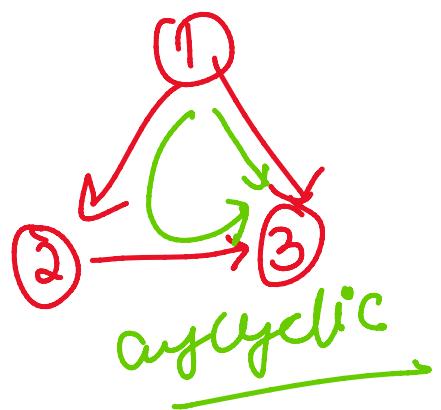
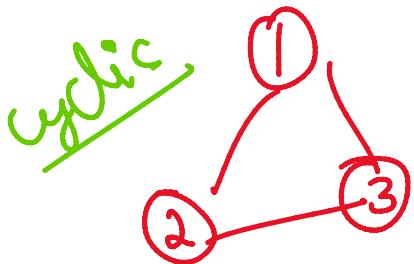
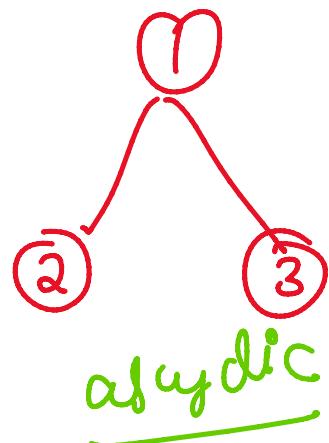
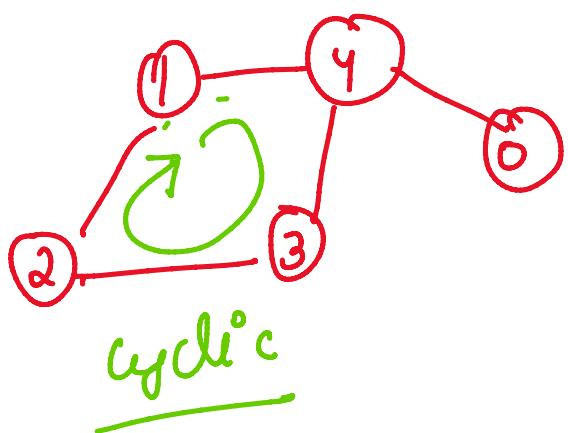


directed

1-2 (2-1)  
2-3  
3-1

② Cyclic / Acyclic

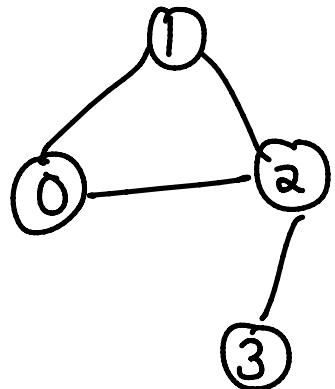
## ② Cyclic/Acyclic



## Graph representation

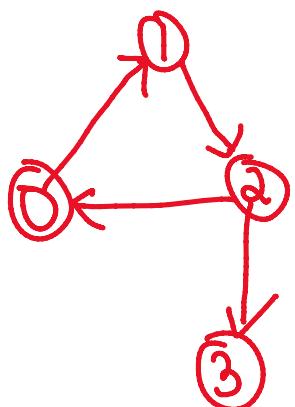
23 September 2023 15:35

Adjacency list +



0 → (1, 2)  
1 → (0, 2)  
2 → (0, 1, 3)  
3 → (2)

vector<int> adj[ ];  
ArrayList<ArrayList<int>>



0 → 1  
1 → 2  
2 → 0, 3  
3 → -

## BFS

23 September 2023 15:39

2 traversals

1. BFS - Breadth first search
2. DFS - Depth first search

(iterative)

(recursive)

BFS

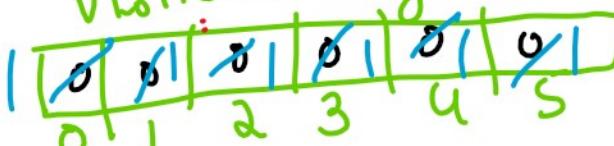
$0 \rightarrow 1, 3$   
 $1 \rightarrow 0, 2$   
 $2 \rightarrow 1, 3$   
 $3 \rightarrow 0, 2, 4$   
 $4 \rightarrow 3, 5$   
 $5 \rightarrow 4$

adj list

Queue



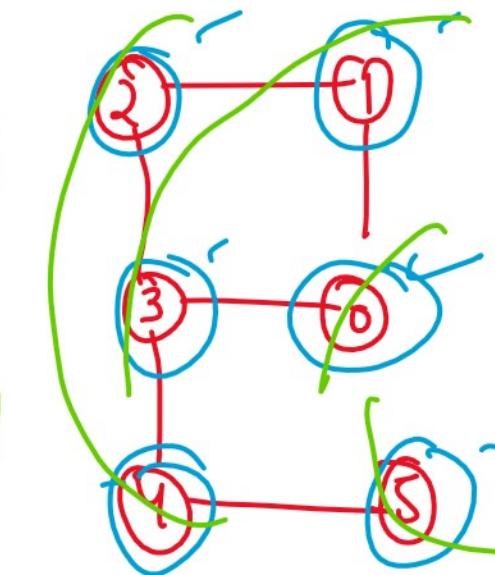
visited array



0, 1, 3, 2, 4, 5



Space  $\sim O(v)$   
Time  $\sim O(v+E)$



DFS

23 September 2023 15:52

# Depth first search (recursive)

adj list:

$0 \rightarrow 1, 3$

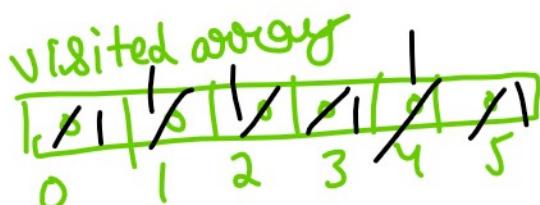
$1 \rightarrow 0, 2$

$2 \rightarrow 1, 3$

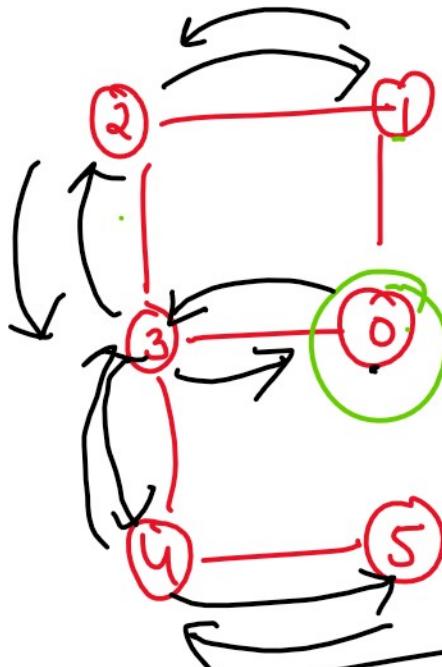
$3 \rightarrow 0, 2, 4$

$4 \rightarrow 3, 5$

$5 \rightarrow 4$



res  $(0, 3, 4, 5, 2, 1)$



Time  $\in O(V + E)$   
Space  $\in O(V)$