BF

TC : O(n log n)
AS : O(1)

Max-Heap.

[3, 5, 4, 2, 9]   K = 3.

TC: O(n log n) + O(K log n)
= O(n log n).

AS : O(n).

3, 5, 4.

[9, 5, 3, 4, 2].   K = 3.

Min-Heap
Top-k.

(9, 5, ~~3~~ 4]

TC: K log(K) + (n-K) log K
= O(n log(K)).

AS: O(K).

arr: [1, 2, 3, 4, 5]   K = 3.

[~~1~~, ~~2~~, 3, 4, 5]

[1, 2, 3, 3, 4, 4]   K=3.

~~1~~, ~~2~~, ~~3~~, 3, 4, 4.

K

0 [→[3, 5, 6, 9]
n →[1, 2, 3, 9]
res [].
→[7, 9, 11, 12]
→[1, 2, 3, 4]].

[1, 2, 3, 3, 5, 6, 9]

K=4   []

[].
[1, 2, 3, 4, 7, 9, 11, 12]

[1, 1, 2, 2

[1, 1, 2, 2, 3, 3, 3, 4, 5, 6, 7, 9, 9, 9, 11, 12]

BF → $O(K^2 \log K^2)$

O

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

0 1 2 3
→ [0, 0, 0, 0]

$$\left[T(K) = \underbrace{T(K/2)}_{\frac{K^2}{2}} + \underbrace{T(K/2)}_{\frac{K^2}{2}} + K^2\right]$$

(0, mid)   0, n-1.
(mid, n-1)
[ ]

---

0 1 K 2 3

0 [→[3, 5, 6, 9]
n 1 →[1, 2, 3, 9]
2 →[7, 9, 11, 12]
3 →[1, 2, 3, 4]].

int.

Min-~~Heap~~

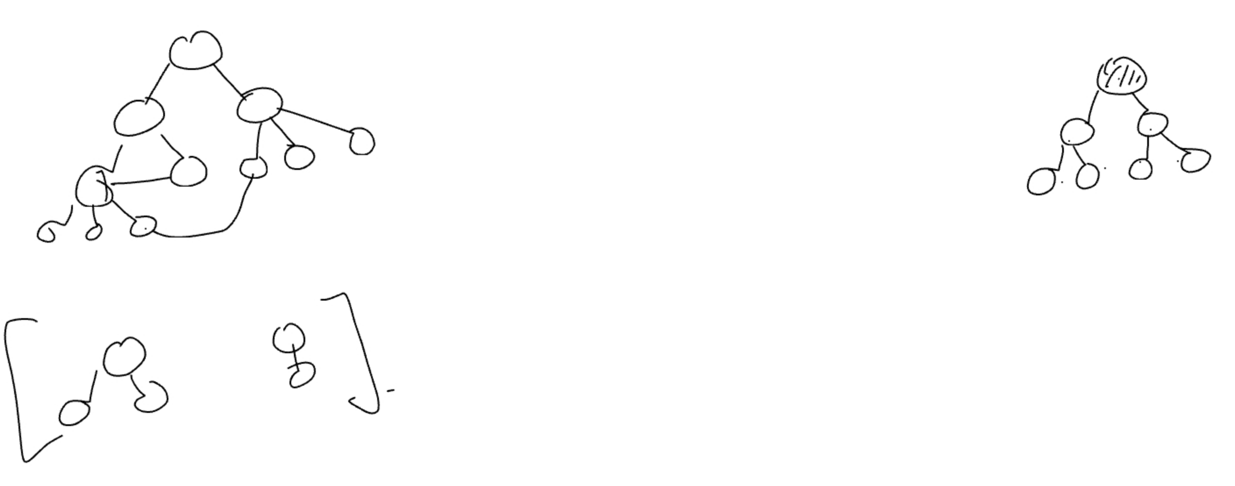
(~~3, 4, 7~~, 1).

{int, [int, int]}
value   x   y.

[1, 1, 2 - - — — -].

TC: O(K² log(K)).
AS: O(K).

Min-Heap
(3, (0,0))
(1, ~~(1,0)~~)
(7, (2,0))
(1, ~~(3,0)~~).
~~(2, (1,1))~~
(2, (3,1))
(3, (1,2))



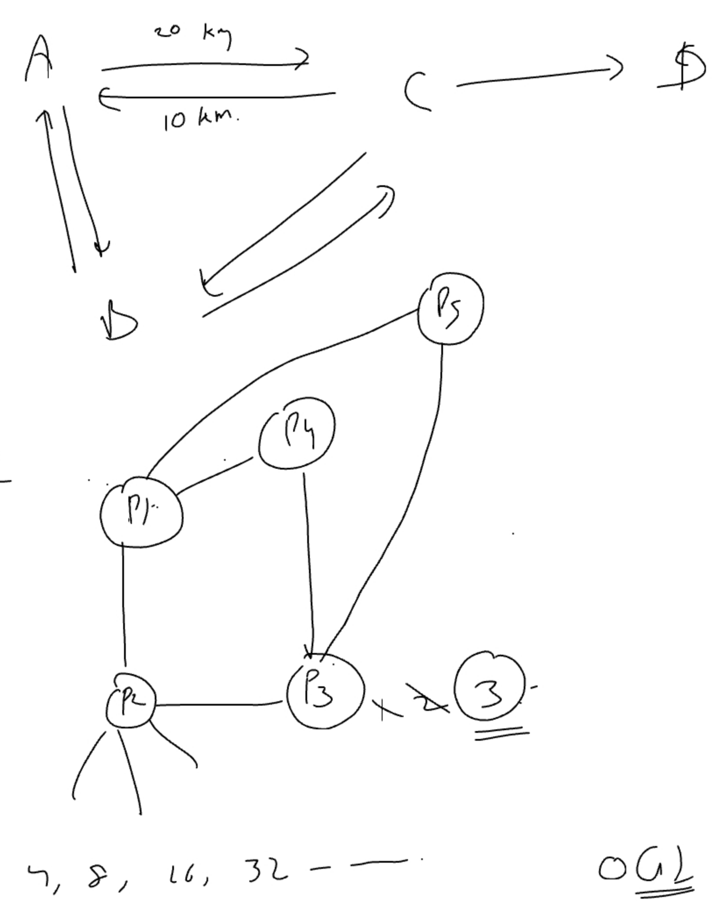
Graph
- No concept of root nodes
- Can be cyclic
- Can have multiple disconnected components

Trees
- Root is present always.
- Are always non-cyclic.
- Have a single connected component.

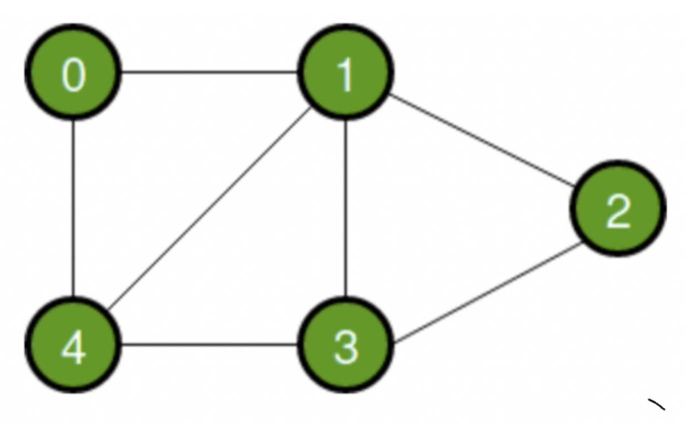
A   B   C.

GPS

Social Network



1, 2, 4, 8, 16, 32 - —   OGL



V = 5.

VXV

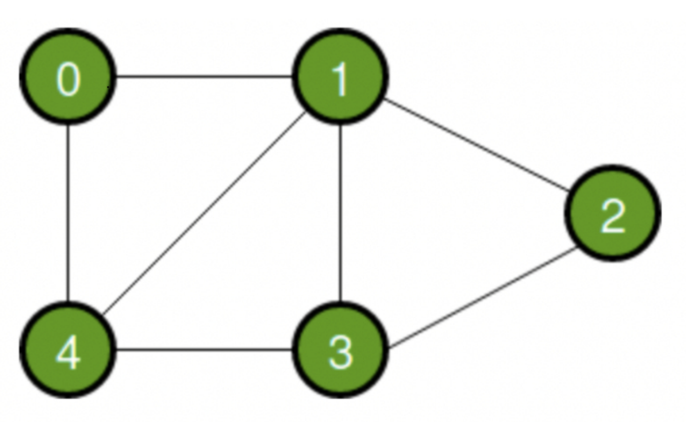
Adj Matrix

	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	1	1
→2	0	1	0	1	0
3	0	1	1	0	1
4	1	1	0	1	0

adj [i][j] = 1
↳ (i → j) edge exists

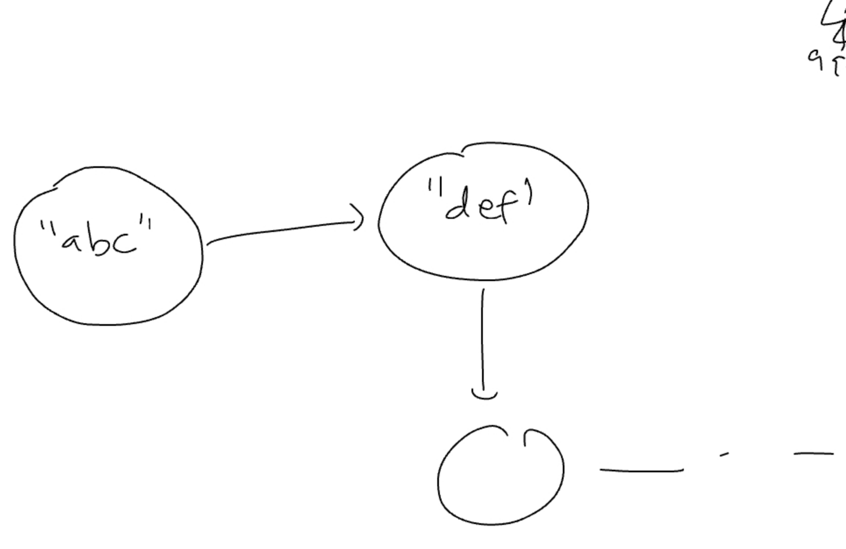
adj [i][j] = 0
↳ (i → j) edge does not exist.

V = 5.   E



0 → [1, 4]
→ 1 → [0, 4, 3, 2]
→ 2 → [1, 3]
3 → [2, 1, 4]
4 → [3, 1, 0].

Space : O(V + E)



Hash Map
"abc" → 0
"def" → 1
⋮
0 → (V-1).

~~DFS~~