* Greedy algorithms continued !!

→ Chocolate Distribution Problem
or Minimum Absolute Difference

[Category → Sliding Window Problem]

arr = [7, 3, 2, 4, 9, 12, 56] $\boxed{m = 3}$

Array Elements = No of chocolates in a packet

Sort →

2	3	4	7	9	12	56
0	1	2	3	4	5	6

$j = m-1$
$= 3-1 = 2$

$M_x - M_n$
4 − 2
7 3
9 4
12 7

answer 56 − 12

* First, last & total occurrence of an element in an array using BinarySearch.

ans = −1

arr = { 1, 2, 3, 3, 3, 3, 4, 5 }

key = 3 fo = 2 lo = 5

f/o

1	2	3	3	3	3	4	5
0	1	2	3	4	5	6	7

TO = lo − fo = 5 − 2 = 3 → +1

$mid = \frac{s+e}{2} = \frac{7}{2} = 3$

arr[mid] == key
* ans = mid
e = mid − 1

arr[mid] < key
s = m+1

arr[mid] > key
e = mid − 1

For FO go to extreme left

key = 3

l/o

1	2	3	3	3	4	5
0	1	2	3	4	5	6

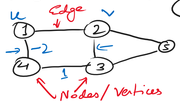
$mid = \frac{s+e}{2} = \frac{7}{2} = 3$

arr[mid] == key
* ans = mid
s = m+1

For LO go to extreme right

Graphs: →

A graph is a non-linear data structure which is used in almost all real-world applications. It has an entity called nodes/vertices which are connected to each other using edges.

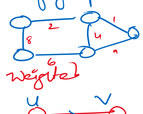


Node → Entity where we store data or value.

Edge → It is the connecting link b/w two nodes/vertex.

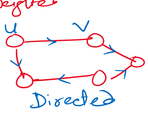
Weights

Types of graphs: →



Weighted

Un-weighted



Directed

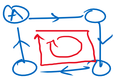
Undirected

u → v = 1
v → u = 1

u → v ✓ = 1
v → u ✗ = 0

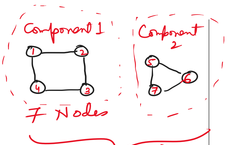
DWG

UWG



cyclic graph

acyclic graph



Disconnected graph

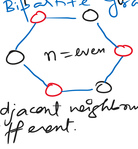
Connected graph

DAG: Directed Acyclic Graph

* Topological Sort → Phase 2

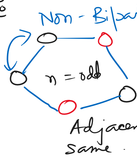
Bipartite Graph:

Bipartite graph



Adjacent neighbours different.

Non-Bipartite



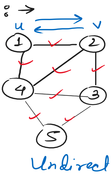
Adjacent neighbors same.

* Graph Representation: →

① Adjacency Matrix: →

vector<vector<int>>

	0	1	2	3	4	5
0						
1		0	1	0	1	0
2		1	0	1	1	0
3		0	1	0	1	1
4		1	1	1	0	1
5		0	0	1	1	0



Undirected