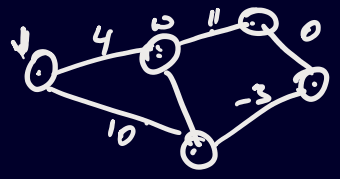
Intro to graph theory

Wednesday, December 22, 2021 11:10 AM

A GRAPH "G" CONSISTS OF A SET OF VERTICIES V & A SET OF EDGES E WHERE EACH EDGE IS A PAIR OF 2 VERTICIES (V, W), WHERE (V, W) ∈ V. SOME TIMES EDGES ARE REFERRED TO AS ARCS. IN SOME CASES EDGES CARRY ADDITIONAL INFORMATION SUCH AS WEIGHT.



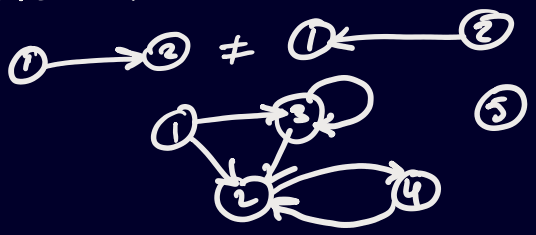
TYPES OF GRAPHS

UNDIRECTED GRAPH

IT'S A TYPE OF A GRAPH WHERE EDGES DON'T HAVE A SPECIFIC DIRECTION. (i.e EDGE (V, W) IS IDENTICAL TO (W, V))

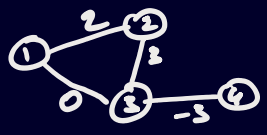
DIRECTED GRAPH (SOMETIMES CALLED A DIAGRAPH)

A DIRECTED GRAPH HAS EDGES WHICH HAVE ORIENTATION $(V, W) \neq (W, V)$



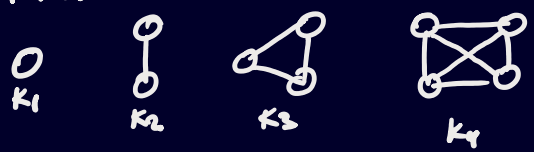
WEIGHTED GRAPH

A GRAPH WHICH INCLUDES WEIGHT INFORMATION ON CONNECTING EDGES AND IT CAN BE APPLIED TO DIRECTED & UNDIRECTED GRAPHS).



COMPLETE GRAPHS

ARE GRAPHS WHERE THERE IS A UNIQUE EDGE CONNECTING EVERY PAIR OF NODES. A COMPLETE GRAPH WITH N VERTICIES IS DENOTED AS GRAPH K_n



DIRECTED ACYCLIC GRAPH [SPECIAL CASE]

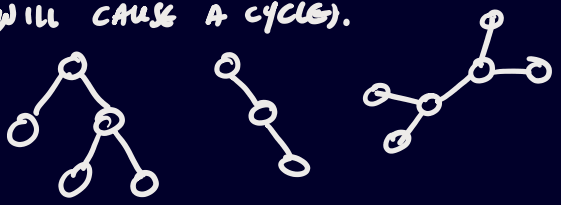
SIMPLY ARE DIRECTED GRAPHS WITH NO CYCLES, IT'S USUALLY

DIRECTED ACYCLIC GRAPH [SPECIAL CASE]

SIMPLY ARE DIRECTED GRAPHS WITH NO CYCLES, IT'S USUALLY USED TO REPRESENT STRUCTURES WITH DEPENDENCIES.

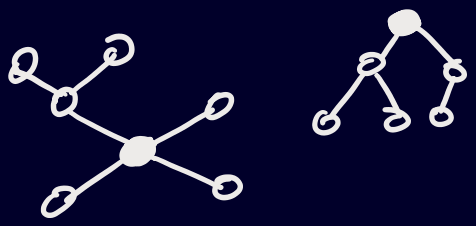
TREE [SPECIAL CASE]

- A TREE IS UNDIRECTED CONNECTED ACYCLIC GRAPH.
- IT IS A GRAPH WHICH MINIMALLY CONNECTED (i.e. REMOVING ONE EDGE MAKES GRAPH DISCONNECTED).
- IT IS A MAXIMALLY CONNECTED GRAPH (i.e. ADDING AT LEAST ONE EDGE TO THE GRAPH WILL CAUSE A CYCLE).



ROOTED TREE

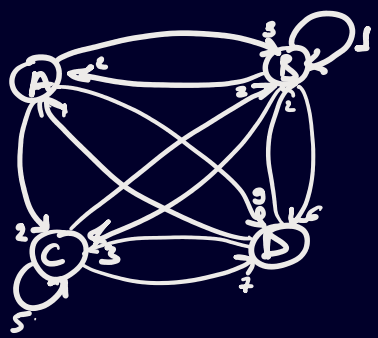
IS A TREE WITH A DEFINED ROOT



GRAPHS DATA STRUCTURE REPRESENTATIONS

ADJACENCY MATRIX

IT'S A VERY SIMPLE WAY OF REPRESENTING A GRAPH. A CELL `M[i][j]` REPRESENTS EDGE WITH A SPESIFIC WEIGHT GOING FROM NODE i TO NODE j



	A	B	C	D
A	0	3	2	9
B	2	1	3	6
C	0	2	5	7
D	1	2	3	0

PROS

- SPACE EFFICIENT FOR REPRESENTING DENSE GRAPHS.

PROS

- SPACE EFFICIENT FOR REPRESENTING DENSE GRAPHS.
- EDGE WEIGHT LOOK UP O(1)
- SIMPLE GRAPH REPRESENTATION.

CONS

- IT REQUIRES $O(V^2)$ SPACE
- ITERATING OVER ALL EDGES TAKES $O(V^2)$
- NOT EFFICIENT WITH SPARSE GRAPHS

ADJACENCY LIST

AN ADJACENCY LIST IS A WAY TO REPRESENT A GRAPH AS A MAP FROM NODES TO LIST OF EDGES.

```
A = [ (B,3), (C,2), (D,9)]
B = [ (A,2), (D,3)]
C = [ (D,1)]
D = [ ]
```

PROS

- SPACE EFFICIENT FOR SPARSE GRAPHS.
- ITERATING OVER ALL EDGES.

CONS

- LESS SPACE EFFICIENT FOR DENSER GRAPHS.
- EDGE WEIGHT LOOK UP IS O[E]

EDGE LIST

IT'S EVEN SIMPLER WAY TO REPRESENT A GRAPH USING UNORDERED LIST OF (u, v, w), IT'S RARELY USED DUE TO IT'S UNSTRUCTURED LAYOUT.

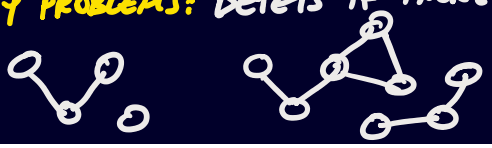
[(A, B, 2), (B, C, 4), (D, A, 0)].

COMMON PROBLEMS SOLVED / ASKED WITH GRAPH THEORY

* **SHORTEST PATH PROBLEM:** FIND SHORTEST BETWEEN 2 POINTS

  ALGORITHMS: BFS, DIJKSTRA, A*, BELMAN-FORD, FLOYD-WARNALL

* **CONNECTIVITY PROBLEMS:** DETETS IF THERE IS A PATH BETWEEN 2 NODES



ALGORITHMS: DFS / BFS, UNION FIND DAT STRUCTURE.

* **TRAVELING SALESMAN**: GIVEN A LIST OF CITIES AND DISTANCES BETWEEN EVERY CITY, FIND THE SHORTEST PATH TO VISIT EACH CITY EXACTLY ONCE AND RETURN TO ORIGIN CITY.

ALGORITHMS: HELD-KARP, ANT COLONY OPTIMIZATION ALGORITHMS. (APPROXIMATING ALGORITHMS)

* **MINIMUM SPANING TREE**: GIVEN A GRAPH IT WILL FIND A TREE THAT CONNECTS ALL VERTICIES WITHOUT ANY CYCLES AND WITH THE MINIMUM COST.

ALGORITHMS: KRUSKAL'S, PRIM'S & BORUVKA'S ALGORITHMS.