## COMP1311 A short intro to graph theory

Г

Andersen Ang J.Southampton UK ngms.science	Basic graph concepts: VVEEMAD Converting multigraph to simple graph Converting digraph to undigraph Subgraph
	Special subgraphs
December 16, 2024	Null & complete graph
	Walk, Trail, Path, Cycle, Circuit
st draft	Connectivity, forest, tree and bipartite
1ay 24, 2023	Tree
	Bipartite
	Combinatorics and graph theory
	Connectivity
	Graph vertex coloring and chromatic polynomial

## The what-why-how of graph

• What is graph: graph theory is not about plotting y = f(x)





not the graph in graph theory

- Why graph: it is a "language" to talk about connectivity
- How to graph: we use set, combinatorics and linear algebra to describe graph
- Theoretical computer science pprox graph theory on steroid
- No panic: we just touch the basic



### Graph isomorphism







• These 3 graphs are the SAME: there is a function F that maps them

$$\begin{array}{c} F(a)=6 \\ F(b)=3 \\ F(c)=5 \\ F(d)=1 \\ F(e)=4 \\ F(f)=2 \end{array} \begin{array}{c} F(u)=3 \\ F(v)=1 \\ F(w)=6 \\ F(w)=6 \\ F(w)=5 \\ F(y)=4 \\ F(z)=2 \end{array} \end{array}$$

• Checking graph isomorphism is generally hard. We don't even know how hard it is (open problem)

#### **Pre-course information**

- What is graph: connectivity structure
- Fancy name of graph: 1-dimensional closure-finite weak topology complex
- Warning: graph theory is VERY hard
- one of the most difficult area in mathematics
- it is universal (can be used in everything)
  - $\implies$  important for computer science
- you probably have never experienced graph theory before
- Study material: lecture slides + workbook + reading books + watch online video

self learning

#### Book

- Discrete Mathematics and Its Applications by Kenneth Rosen
- A First Look at Graph Theory by John Clark and Derek Allan Holton
- Introduction to graph theory by Douglas West
- Graph Theory by Reinhard Diestel
- Schaum's Outline of Graph Theory
- Outcome: understand the very basics of graph

Prerequisite

Set

Matrix

Combinatorics

enough for this course first 41 pages

free but not for first reading for more practise problems

### Motivation by application of graph theory

Image segmentation by graph cut
http://cs-www.cs.yale.edu/homes/spielman/sgta/

## Contents

#### Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

## VVEEMAD

• Definition A graph G(V, E) has a vertex set V and an edge set E.

- V is the set of vertices. Here  $V = \{1, 2, 3, 4, 5, 6\}$ . |V| = the number of vertex in V = the cardinality of V. Here |V| = 6.
- E is the set of edge connecting a pair of vertice  $i, j \in V$ .
- $1 \rightarrow 1$  so we have an edge (1,1)
- $1 \rightarrow 2$  so we have an edge (1,2)
- $1 \rightarrow 3$  so we have an edge (1,3)
- $1 \rightarrow 5$  so we have an edge (1,5)
- $3 \rightarrow 4$  so we have an edge (3, 4)
- $4 \rightarrow 1$  so we have an edge (4, 1)
- $4 \rightarrow 3$  so we have an edge (4,3)
- $5 \rightarrow 3$  so we have an edge  $(5,3)_a$
- $5 \rightarrow 3$  so we have an edge  $(5,3)_b$
- $5 \rightarrow 4$  so we have an edge (5,4)

 $E = \left\{ (1,1), (1,2), (1,3), (1,5), (3,4), (4,1), (4,3), (5,3)_a, (5,3)_b, (5,4) \right\}. |E| = 10.$ 

- Terminology
  - Vertex = node = dots = points
  - Edge = arc = curve = line
  - Two edges sharing the same vertices are *parallel*, e.g.,  $(5,3)_a$  and  $(5,3)_b$ .
  - (1,1) is a self-loop



$$\mathbf{M} \qquad E = \left\{ (1,1), (1,2), (1,3), (1,5), (3,4), (4,1), (4,3), (5,3)_a, (5,3)_b, (5,4) \right\}$$

• From E, we get incidence matrix M



• *M* expressed as an **incidence list** (code-friendly)

**A**  

$$E = \left\{ (1,1), (1,2), (1,3), (1,5), (3,4), (4,1), (4,3), (5,3)_a, (5,3)_b, (5,4) \right\}$$
• From *E* we get adjacency matrix *A*

$$A \in \{0, 1, 2, ..., |V|\}^{|V| \times |V|}, \text{ where } [A]_{ij} = \begin{cases} 0 & \text{if } (i, j) \text{ and } (j, i) \notin E \\ 1 & \text{if } (i, j) \text{ or } (j, i) \in E \\ 2 & \text{if two } (i, j) \text{ or } (j, i) \in E \\ \vdots \end{cases}$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad A = A^{\top} \quad \begin{array}{c} \text{space (storage) complexity:} \mathcal{O}(|V|^2) \\ \text{search (time) complexity:} \mathcal{O}(1) \\ \end{array}$$

• A expressed by **adjacency list** (code-friendly)





$$M = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

• In-degree: number of edges come to the node

n

- Out-degree: number of edges leave from the node
- Degree: number of edges touching the node



0

#### First Theorem of Graph theory: Handshaking lemma

• Theorem (Euler 1736) For any graph G with |E| edges and |V| = n vertices

The sum of degree of all vertices 
$$=\sum_{i=1}^{|V|} \deg_i = 2|E|,$$

where  $\deg_i$  stands for degree of vertex *i*.

**Proof** Each edge has two end vertices, thus contributes exactly 2 to the sum of the degrees.

#### • Handshaking interpretation

In a party of n people, the total number of handshakes equals to 2 times the number of handshaked pairs.

• Pigeonhole principle.

## All graph has even number of odd vertices

• 2

- Definition An vertex is called odd (even) if its degree is odd (even).
- **Corollary** In any graph, there is an even number of odd vertices.

Proof
$$\sum_{i=1}^{|V|} \deg_i = 2|E|$$
Hanshaking lemma
$$\iff \sum_{i \in \text{odd}} \deg_i + \sum_{i \in \text{even}} \deg_i = 2|E|$$
split vertices into odd and even group
$$\iff \sum_{i \in \text{odd}} \deg_i = 2|E| - \sum_{i \in \text{even}} \deg_i$$
$$2|E| - \sum_{i \in \text{even}} \deg_i \text{ is an even number}$$
$$2|E| is even$$
$$\sum_{i \in \text{even}} \deg_i \text{ is even because all the vertex } i \text{ here has even degree}$$
For
$$\sum_{i \in \text{even}} \deg_i \text{ to be even, there must be even number of odd vertices.}$$

13/71

## We ignore isolated node(s)

- Graph theory is about interaction, no edge = no interaction, so we ignore isolated node



• Graphs that all vertices are isolated are called **null graph**, denoted by N

## Directed graph (digraph) and undirected graph (undigraph)

- Directed graph: have arrow
- Undirected graph: no arrow
- Undirected graph is not the same as bidirected graph (out of scope)





## Simple graph, multigraph and pseudograph

	Pseudograph	Multigraph	Simple graph
have self loop	ok	no	no
have multiedge	ok	ok	no

- multiedge: edge connecting the same pair
- self loop: edge connecting the same node
- Multigraph can be converted to simple graph  $\implies$  we focus on simple graph



#### Exercise: draw the simple graph

• G(V, E) with  $V = \{1, 2, 3, 4\}$  and  $E = \{(1, 2), (2, 3), (2, 4)\}$ 

• G(V, E) with  $V = \{1, 2, 3, 4, 5\}$  and  $E = \{(1, 2), (2, 3), (3, 1), (4, 5)\}$ 

• G(V, E) with  $V = \{1, 2, 3\}$  and  $E = \{\}$ 

Exercise: find the  ${\cal G}(V,E)$  for these unlabeled simple undigraphs



Exercise: find the  ${\cal G}(V,E)$  for these unlabeled simple digraphs



Ok now go turbo on practice: write down the VVEEMAD



20/71

## Contents

#### Basic graph concepts: VVEEMAD

#### Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

#### Converting multigraph to simple graph



## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

#### Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

## Draw a graph from ${\cal M}$

• Consider a directed graph 
$$G(V, E)$$
 with  $M = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ .

- The draw the graph from the given information
- From the number of rows in M, we know that there are 3 vertices, |V| = 3. Suppose we name the 3 vertices x, y, z.

$$V = \{x, y, z\}, |V| = 3.$$

- $\bullet~\mbox{From}~M$  we know the arrows
- $M_{1,2} = 1$  means  $x \to y$
- $M_{2,1} = 1$  means  $y \to x$
- $M_{2,3} = 1$  means  $y \to z$ •  $M_{3,1} = 1$  means  $z \to x$

- $E = \{(x, y), (y, x), (y, z), (z, x)\}$
- 4 non-zeros in M means |E| = 4.



# Converting digraph to undigraph

- Consider the edge e(x, y)
- Replace e(x, y) by  $(x, v_1), (v_1, v_2), (v_1, v_3), (v_3, v_4), (v_4, v_5), (v_3, y)$



- To go back from undigraph to digraph
  - Identify all leaf (degree-1 vertex)
  - Find the leaf whose neighbour has degree 2
  - The neighbour is  $v_4$  and it has neighbour  $v_3$
  - $v_3$  has a unique neighbour that
    - has degree 3, and
    - adjacent to a leaf
    - (this neighbour of  $v_3$  is  $v_1$  and  $v_1$  is adjacent to a leaf  $v_2$ )
  - The other neighbour of  $v_1$  is x
  - The other neighbour of  $v_3$  is y
  - Delete  $v_1, v_2, ..., v_5$  and connect arrow from x to y

 $\{x, v_2, v_5, y\}$  $\{v_5\}$  because  $\deg(v_4) = 2$ 



## Summary

- $\bullet \ V, |V|, E, |E|, M, A, D$
- Direct, undirect, multiedge, self-loop, simple
- Converting multigraph to simple graph
- Converting digraph to undigraph

not in exam

not in exam

# From now on we focus on simple undigraph.

## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

#### Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

## Subgraph

• Like subset in set, we can define subgraph of a graph

A subgraph of a graph G(V, E), is a graph S(U, F) that  $U \subseteq V, F \subseteq E$ .

 ${\cal S}$  can be obtained from  ${\cal G}$  by deleting edges and/or vertices.

- Trivial fact: every graph is a subgraph of itself.
- Example



- Set operations carry over to graph
  - Intersection
  - Union
  - Complement

# Subgraph

• A graph G(V, E) with |V| = 5, |E| = 8





- Subgraph is not unique: there are 6 possible triangles T from G
- This is NOT a triangle because it has four vertices



## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

Two extreme simple undigraph: null and complete graphs

- $V = \{1, 2, 3, 4\}$ , |V| = 4
- $E = \varnothing$ , |E| = 0
- M = A = D = 0
- Null graphs are not interesting





• Complete graph of n vertices are denoted by  ${\cal K}_n$ 

#### Two extreme simple graphs: null and complete graphs

- Consider simple graph G with |V| = n.
- Null graph  $N_n$  is G with the smallest possible |E|
- **Complete graph**  $K_n$  is G with the largest possible |E|
- A complete graph is a graph in which all pair of vertices is joined by an edge
- $K_n$  for n=1 to 8



- All graphs with n vertices are between  $N_n$  and  $K_n$ , and we can define sparse and dense graph as
- A graph is sparse if  $|E| \ll O\left(\frac{|V|(|V|-1)}{2}\right) = O(|V|^2)$
- A graph is *dense* if  $|E| \approx \mathcal{O}(|V|^2)$
- Recall Big-O notation  $f(x) = \mathcal{O}(q(x))$  if  $f(x) \le Mq(x)$  for sufficiently large x

|E| = 0

 $|E| = \frac{|V|(|V| - 1)}{2} = \binom{|V|}{2}$ 

### Walk, trail, path, cycle

- Definition A walk is a sequence  $v_0 \rightarrow v_2 \rightarrow \ldots v_m$  in a graph.
  - $v_0$ : initial vertex/ source
  - $v_m$ : final vertex/ sink
- The number of edges in a walk, m+1, is called its length
- Example  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 5 \rightarrow 4 \rightarrow 2$  is a length-7 walk. Walk can also be expressed using sequence of edge

$$W = \left\{ (1,2), (2,3), (3,4), (4,5), (5,5), (5,4), (4,2) \right\}$$

We say this walk  $\ensuremath{\textit{traverses}}$  the edges in W

• Definition A trail is a walk if distinct edges.

e.g.  $\left\{(1,2),(2,3),(3,4),(4,5),(5,5)\right\}$ 

Another definition of trail: a walk that traverse each edge at most once.

• Definition A path is a trail if distinct vertices except possibly source = sink e.g.  $\{(1,2),(2,3),(3,4),(4,5)\}$ 

Another definition of path: a trail that traverse each vertex at most once.

• Definition A cycle is a closed path (source = sink) e.g.  $\{(1,3),(3,4),(4,1)\}$ 



	walk	trail	path
edge repeat	ok	no	no
vertex repeat	ok	ok	no

### Example of path graph and cycle graph

• A path graph with length- $\ell$  is a graph with  $V = \{1, 2, .., \ell\}$  and  $E = \{(1, 2), (2, 3), ..., (\ell, \ell + 1)\}$ , after renaming the vertices and edges Path graphs of length  $\ell = 1, 2, 3, 4$ 

• A cycle of length  $\ell \ge 3$  is a graph with  $V = \{1, 2, .., \ell\}$  and  $E = \{(1, 2), (2, 3), ..., (\ell, 1)\}$ , after renaming the vertices and edges

Cycle graph of length  $\ell = 3, 4, 5, 6$ 



## Other structures: circuit (= a closed trail)

#### not in exam

- Circuit vs cycle: disregard starting and ending vertices
  - Circuit allows repeated vertices
  - Cycle does not allow repeated vertices
- Eulerian: a circuit consists of a closed path that visits every edge of a graph exactly once "use each road exactly once, possibly visiting same city many times"
- Hamiltonian: a circuit that visits every vertex of a graph exactly once. "visits each city exactly once, possibly using the same road many times" A circuit



• Chinese postman problem (Meigu Guan, 1960): find a shortest circuit that visits every edge at least once

### Other structures: clique

#### not in exam

- Clique: a set C of G that all pair of distinct vertices in C are adjacent.
  - the subgraph induced by a clique is a complete graph
  - maximal clique: a clique that cannot be made larger by adding more vertices from the graph



#### Other terms

#### not in exam

Boundary Block Claw Chord DAG expander face induced matroid perfect plannar Ramanuian space total successor weight

Surface area Embedding degeneracy intersection neighbour reachable spanner wealkly connected

Diameter Star circumference depth forbidden list network power rectangle split treewidth wheel

Eccentricity Butterfly core diamond height minor order proper saturated square utilitv

Bandwidth Chain cut dual hole modular orientation quasi sibling strong unweighted



https://mathworld.wolfram.com/PetersenGraph.html

**Other structures:** *k***-regular** 

not in exam



#### Connectivity, forest, tree and bipartite



- A graph is connected if for any  $i \in V, j \in V, i \neq j$ , there is a path connecting i to j.
- Definition A forest is an acyclic graph. Acyclic = no cycle
- **Definition** A **tree** is an acyclic connected graph.
- **Definition** A **bipartite** is graph that vertices can be divided into two parts such that there is no edges within each part.

Acvclic = no cvcle

How to tell a graph is bipartite

- Theorems
- Every tree is bipartite
- A graph is bipartite if and only if it has no subgraph that has an odd-length cycle.
- Application of bipartite: assignment problem, stable marriage problem



## Summary

		repeated vertices	repeated edge	open/closed	
-	Walk	Y	Y	Both	
	Trail	Y	Ν	0	
	Path	N	Ν	0	
	Cycle	N	Ν	С	
	Circuit	Y	Ν	С	
Term	definition				
Connected		there is a path for any $(i, j)$			
Forest	acyclic graph				
Tree	acyclic connected graph				
Bipartite	vertices can be divised into two parts with no edges within each part				

### Do you remember these terms?

Vertex	Edge		
Adjacency	Degree	Directed	Undirected
Multigraph	Pseudograph		
Complete	Subgraph		
Walk	Trail	Path	Cycle
Clique			
Tree	Bipartite		
lsolated			
	Vertex Adjacency Multigraph Complete Walk Clique Tree Isolated	VertexEdgeAdjacencyDegreeMultigraphPseudographCompleteSubgraphWalkTrailCliqueFreeTreeBipartiteIsolatedFree	VertexEdgeAdjacencyDegreeDirectedMultigraphPseudographCompleteSubgraphWalkTrailPathCliqueTreeBipartiteIsolatedValkValk

gray = ok to ignore

## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

#### Tree

Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

#### **Tree and Forest**

Tree = acyclic connected graph



6 vertices, 5 edges, 1 tree



7 vertices, 5 edges, 2 trees

- Null graph  $N_n$  is also a forest (the whole forest has no edge)
- Equivalence conditions of tree

 $\boldsymbol{G}$  is connected and acyclic

- $\iff G$  is connected, and become disconnected if any single edge is removed from G
- $\iff G$  is acyclic, and become cyclic if any single edge is added to G
- $\iff$  any two vertices in G can be connected by a unique simple path

## Tree terminology

- The top vertex is called **root**
- A is the **parent** of A1 and A2
- A1 and A2 is the **children** of A
- A, B are branch nodes
- A1, A2, B1, C are leaf nodes
- **Height** := the length from root to the furthest vertex The height is 3 here
- The **depth** of A is 1 and the depth of A1 is 2.
- At depth-0, the width is 1. At depth-1, the width is 3
- A subtree is you chop a branch out of the main tree

### not in exam



## **Application of Tree**

- Mental picture for combinatorics
- Syntax tree
- BFS
- DFS
- Spanning Tree
  - Prim
  - Kruskal
- Data structure
  - binary tree
  - red-black tree
- Adelson-Velsky-Landis
- Stack (Last-in-First-out)
- Queue (First-in-First-out)

### not in exam

#### Monte Carlo Tree Search (AlphaGo)



## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

#### Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

## **Bipartite**

- A graph is bipartite if its vertices can be renamed into two groups  $\{L_1, L_2, ..., L_i, R_1, R_2, ..., R_j\}$  and each of its edge joins  $i \in L$  and  $j \in R$ .
- L,R stands for left and right
- A bipartite graph is that there is no edge within the L and there is no edge within the R
- E.g. Six bipartite graphs



- A complete bipartite is a bipartite that every vertex in L is joined by an edge to every vertex in R
- In the 6 graphs

not complete not complete not complete not complete not complete complete The last graph is complete bipartite: all  $L_i$  is joined to every  $R_j$ 

## It is not trivial to see if a graph is bipartite



• But actually it is bipartite



Is this bipartite?
 This is not bipartite.
 How do you know?

### How to decide a graph is not bipartite

• Theorem A graph is bipartite if it contains no odd-cycle.

• Check for Odd-Length Cycles

A graph is bipartite if and only if it does not contain any odd-length cycles in its subgraphs. If an odd-length cycle is found in the graph, it is not bipartite.

• Graph-coloring and Breadth-First Search (BFS)

Perform a BFS starting from any vertex.

Assign the starting vertex one color, alternate colors for each level of the BFS tree.

If two adjacent vertices with the same color is found ing this process, the graph is not bipartite.

## Understanding the no odd-cycle theorem of bipartite



- Is this bipartite? 4—3
   Method 1: By the "No odd-cycle Theorem" (there is a 5-cycle), this graph is not bipartite
- Method 2: we draw it to see it 1. Write vertex-1 on the right

1

2. Write all vertices adjacent to vertex-1 to the left



3. Bipartite means "left-right", for the remaining vertices, we put them to using a left-right-left-right order.

For vertex-2, its adjacent vertex-3, put to the right



4. For vertex-5, its adjacent vertex-4, put to the right



5. For vertex-4, connect the remaning edge



The (3,4) is making the graph not bipartite

Algorithm 1: IsGBipartite: a naive Bipartite checking algorithm

**Input:** A simple undigraph G(V, E)**Output:** True if G is bipartite, False otherwise Initialize  $L = \emptyset$ ,  $R = \emptyset$ ,  $Q = \emptyset$ // left-set, right-set, queue 2  $L = \{1\}, Q = \{1\}$ // add node 1 to L and enqueue 1 to Q3 while  $Q \neq \emptyset$  do Take a node  $i \in Q$ for each neighbor i of i do 5 if *j* is not visited then 6 if  $i \in L$  then 7  $R = R \cup \{j\}$ 8 // add i to Lelse Q  $L = L \cup \{j\}$ // add j to R10  $Q = Q \cup \{j\}$ // enqueue j to Q11 12 for each  $(i, j) \in E$  do if  $i \in L$  and  $j \in L$  or  $i \in R$  and  $j \in R$  then 13 return False 14 15 else 16 return True

How fast is it? Can we make it faster? What is the theoretical fastest way to do this? Is there other way?

## **Application of bipartite**

#### Application

- Job-assignment problem in resource Allocation Slide 18 https://angms.science/doc/teaching/ CO327/OIntro\_motivation\_examples.pdf
- Stable marriage problem
- Recommendation system
- Social Network
- Transportation Networks

#### Example of real-life bipartite

- Movie-Cast Networks for telling is a new movie good based on historical linking data
- Customer-Product Purchases for advertisement
- Drug-Target Interaction represent relationships between drugs and molecular targets in pharmacology
- user-music playlist generation create personalized music playlists

## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

#### Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

#### How many k-cycles in a simple complete graph $K_n$ ?

**Question**: how many 3-cycle (triangle) in  $K_8$ ?

Theorem  
Proof  
There are 
$$\frac{1}{2k} \frac{n!}{(n-k)!} k$$
-cycles in  $K_n$ .  
 $\begin{pmatrix} n \\ k \end{pmatrix}$  number of ways to choose  $k$  vertices among  $n$  vertices  
 $(k-1)!$  the number of orderings in the selected  $k$ -set  
2 number of orientation of the cycle (clockwise and anticlockwise)  
 $\frac{\binom{n}{k}(k-1)!}{2}$  by product rule and division rule  
 $\frac{\binom{n}{k}(k-1)!}{2} = \frac{\frac{n!}{(n-k)!k!}(k-1)!}{2} = \frac{1}{2k}\frac{n!}{(n-k)!} = \frac{1}{2k}n^k$   
There are  $\frac{1}{2\cdot 3}\frac{8!}{(8-3)!} = 56$  triangles in  $K_8$ . Similarly, there are  $\frac{1}{2\cdot 3}\frac{4!}{(4-3)!} = 4$  triangles in  $K_4$ .  
 $57/71$ 

### Combinatorics of graph can be very difficult

• Question: What is the maximum number of edges in a triangle-free *n*-vertex graph? Answer (Mantel's Theorem)  $\lfloor \frac{n^2}{4} \rfloor$ So for 5-vertex graph, it is possible to have a 6-edge, can you draw it?

• Question: What is the minimum number of triangle in a *n*-vertex *m*-edge graph? Answer  $\frac{4m}{3n}\left(m-\frac{n^2}{4}\right)$ 

- (Odd Town Problem) A town with n citizen has m clubs such that
- each club has an odd number of members
- any two different club share an even number of common members

For the conditions to hold, we must have  $m \leq n$ 

### **Combinatorics of bipartite**

- E.g. In a (m, n) bipartite graph, what is the smallest possible number of edges and the largest possible number of edges?
  - Smallest possible: null graph, so |E| = 0
  - Largest possible: complete bipartite, so |E| = mn
- E.g. In a bipartite graph G(V, E) with n = |V|, what is the largest possible number of edges?
- The maximum number of edges occurs when the vertices are divided as evenly as possible into two sets L, R.
- If n is even, each set will have  $\frac{n}{2}$  vertices.
- If n is odd, one set will have  $\left\lfloor \frac{n}{2} \right\rfloor$  vertices and the other will have  $\left\lfloor \frac{n}{2} \right\rfloor$  vertices.
- Mximum number of edges is:

$$\left\lfloor \frac{n}{2} \right\rfloor \times \left\lceil \frac{n}{2} \right\rceil$$

## **Combinatorics of tree**

- Theorem A tree T(V, E) with n vertices has n 1 edges Proof Mathematical induction with strong induction
- Theorem A tree T(V, E) with n vertices has  $n^{n-2}$  ways to label the vertices.

#	Root	Left Child	Right Child	Left Grandchild
1	1	2	3	4
2	1	2	4	3
3	1	3	2	4
4	1	3	4	2
5	1	4	2	3
6	1	4	3	2
7	2	1	3	4
8	2	1	4	3
9	2	3	1	4
10	2	3	4	1
11	2	4	1	3
12	2	4	3	1
13	3	1	2	4
14	3	1	4	2
15	3	2	1	4
16	3	2	4	1

60 / 71



## Combinatorics of hypercube graph $Q_n$





- **Theorem**  $Q_n$  has  $2^n$  vertices
  - Each vertex in  $Q_n$  can be represented by an *n*-bit binary string.
  - As each bit can be either 0 or 1, there are  $2^n$  possible combinations of these bits
- **Theorem**  $Q_n$  has  $n2^{n-1}$  edges
- Each vertex in  $Q_n$  is connected to n other vertices: flipping any one of the n bits in the binary representation of a vertex will result in an adjacent vertex.
- As there are  $2^n$  vertices and each vertex has n edges, there are  $n \cdot 2^n$  edges, but this double counted each edge, half of this gives  $n2^{n-1}$ .
- Theorem  $Q_n$  has  $2^{2^n-n-1} \prod_{k=1}^n k^{\binom{n}{k}}$  spanning trees (WTF?)

You can see this number grows very quickly to a large number, but in fact this is still a "small number"

# Combinatorics of graph can go crazily large: Kirby-Paris Hydra

- Start with a tree (hydra) In each turn n (n is the turn number), the player pick a vertex h
  - h means head, it can only be a leaf in the tree
  - let  $p = \operatorname{Parent}(h)$  and  $g = \operatorname{Parent}(p)$
  - chop head: remove h from the tree
  - grows n additional copies of modified p as children on g



- E.g. Hydra(n) = number of steps required to chop a head of depth n with no further right branches
   n
   1
   2
   3
   4
   Hydra(n)
   1
   3
   37
   big number ≫ number of atoms in the observable universe
   10<sup>82</sup>
- How do you study big number: ordinal  $\omega, \ \omega + 1, \ \omega^2, \ \omega^{\omega}$ ,  $\log, \log \log$
- Googology Wiki

### Combinatorics, graph theory and matrix walk into a bar ...

- First Theorem in Algebraic Graph Theory. Given the adjacent matrix of a graph G(V, E). The number of length-k walks starting from vertex i to vertex j is  $(A^k)_{ij}$ .
- Proof by mathematical induction
- Base case: for k = 1,  $A_{ij}^k = A_{ij}$  is the number of length-1 walk from i to j
- Hypothesis case: assume the statement is true at case k = n.
   I.e., the number of length-k walks starting from vertex i to vertex j is (A<sup>k</sup>)<sub>ij</sub>.
- Inductive step: consider the case k = n + 1.
  - Consider  $A^{n+1} = A^n A$ .
  - Now the number of length-(n + 1) walks between i to j equals the number of length-n walks from i to v that is adjacent to j, which is the (i, j) entry of  $A^n A = A^{n+1}$  the non-zero entries of the column of A corresponding to v are precisely the first neighbours of v.

E.g. How length-3 walks from 5 to 3?



64/71

## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

Combinatorics and graph theory

### Connectivity

Graph vertex coloring and chromatic polynomial

## Connectivity

- A graph is connected if for any  $i, j \in V$ , the graph G has a subgraph that is a path from i to j
- **Theorem** All connected graph with no subgraph that is odd-cycle is bipartite Proof: mathematical induction
- Bridge A bridge in a connected graph G is an edge whose removal disconnects G



- Fact: for a path graph, every edge is a bridge.
- Fact: for any connected graph, it contains a spanning tree

## Algebraic connectivity: Laplacian matrix

• Laplacian = Degree - Adjacency

$$L = D - A$$

• Theorem G(V, E) is a connected graph  $\iff$  the second smallest eigenvalue of L(G) is larger than zero

$$\begin{array}{c} \begin{array}{c} \mbox{Label} \\ \mbox{Root: 1} \\ \mbox{Left Child: 2} \\ \mbox{Right Child: 3} \\ \mbox{Left Grandchild: 4} \end{array} \\ \begin{array}{c} \mbox{Label} \\ \mbox{Right Child: 3} \\ \mbox{Left Grandchild: 4} \end{array} \\ \begin{array}{c} \mbox{Label} \\ \mbox{Left Grandchild: 4} \end{array} \\ \begin{array}{c} \mbox{Label} \\ \mbox{Labe$$

• This is similar to you using  $[x^n]$  of a polynomial to count things in combinatorics. You previously learned how to count things using polynomial of x, a scalar. Now in graph theory, we count things using polynomial of matrix

## Contents

Basic graph concepts: VVEEMAD

Converting multigraph to simple graph

Converting digraph to undigraph

Subgraph

Special subgraphs Null & complete graph Walk, Trail, Path, Cycle, Circuit Connectivity, forest, tree and bipartite

Tree

Bipartite

Combinatorics and graph theory

Connectivity

Graph vertex coloring and chromatic polynomial

### Graph vertex coloring and chromatic polynomial

- Proper coloring: two adjacent vertices with different color
- Chromatic polynomial: the number of ways a graph can be properly colored
- Notation:  $\chi_G(t)$  is the chromatic polynomial of a graph G, here t is the number of color you can use
- Line  $L_2$ ,  $\chi_{L_2}(t) = t(t-1) = t^2 t$
- Line  $L_3$ ,  $\chi_{L_3}(t) = t(t-1)^2 = t^3 2t^2 + t$
- Line  $L_n$ ,  $\chi_{L_n}(t) = t(t-1)^{n-1}$
- Triangle  $K_3$ ,  $\chi_{K_3}(t) = t(t-1)(t-2) = t^3 3t^2 + 2t$
- Square  $\chi_{square}(t) = t^4 4t^3 + 6t^2 3t = t(t-1)^2 + t(t-1)(t-2)^2$ 
  - For vertex *a*, you have *t* ways to color



• For vertices b and d, you have t-1 ways to color case 1. color of b = color of dThen for c you have t-1 ways to color case 2. color of  $b \neq \text{color of } d$ Then for c you have t-2 ways to color



• What's the big deal: you can use a polynomial to represent a graph !!!!

W/TF?7

## Other topics

- Graph complement, graph disjoint, graph intersection, graph union
- Weighted graph, Graph cut, graph flow
- Eulerian Graph, Hamiltonian path, Petersen graph, Ramanujan graph
- Graph Laplacian L = D A
- Graph theory + Linear algebra gives
- Spectral graph theory
- Matroid

view graph as matrix, use eigendecomposition to study graph abstraction based on the notion of linear independence

Graph algorithms you will learn next semester / next year

- Dijkstra's alg.
- Bellman-Ford alg.
- Floyd-Warshall alg.
- Prim's alg.
- Kruskal's alg.
- Ford-Fulkerson alg.

Hypergraph and topology

finding shortest path of weighted graph generalized Dijkstra dynamic programming for finding shortest path of weighted graph node-based, finding min. spanning tree edge-based, finding min. spanning tree max-flow

## Summary

- $\bullet \ V, |V|, E, |E|, M, A, D$
- Direct, undirect, multiedge, self loop, simple
- Converting multigraph to simple graph
- Walk, trail, path, circuit, cycle, forest, tree, bipartite
- Number of walks
- Connectivity
- Coloring and chromatic polynomial