Plan:

- Matrix perspectives on walks & circuits
- Back to minimal spanning trees
- Start connectivity (Menger's theorem)

Matrix representation:

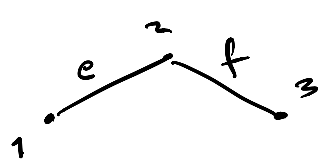
vertices numbered $1, \ldots, n$ $\longleftrightarrow$ indexing entries in vectors/matrices

associated to a (simple) graph G, get a matrix

$$T_G \qquad (T_G)_{i,j} = \begin{cases} 0 & \text{if } i \,\&\, j \text{ are not adjacent} \\ 1 & \text{if } i \,\&\, j \text{ are adjacent} \end{cases}$$

$$T_G^2 \qquad T_G^3$$

Proposition: $(T_G{}^k)_{i,j}$ = # of walks of length k from i to j.



$$\begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 1_e & 0 \\ 1_e & 0 & 1_f \\ 0 & 1_f & 0 \end{bmatrix} \end{array}$$

$$\begin{bmatrix} 0 & 1_e & 0 \\ 1_e & 0 & 1_f \\ 0 & 1_f & 0 \end{bmatrix} \begin{bmatrix} 0 & 1_e & 0 \\ 1_e & 0 & 1_f \\ 0 & 1_f & 0 \end{bmatrix} = \begin{bmatrix} 1_e^2 & 0 & 1_e 1_f \\ 0 & 1_e^2 + 1_f^2 & 0 \\ 1_f 1_e & 0 & 1_f^2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Observations:

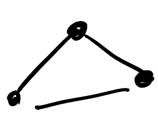
$\mathrm{tr}(T_G^2)$ = # closed walks of length 2.

$= 2(\#\text{edges}) = \sum \deg(v)$

degree formula.

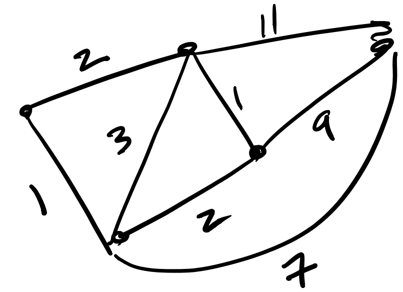
$\mathrm{tr}(T_G^3)$ = # closed walks of length 3.

(all come from $\triangle$'s)



$= 6 \cdot \underbrace{\#\text{triangles}}_{\text{cycles subgraphs of length 3}}$ in G.

Back to
Minimal spanning trees



Prim's Algorithm

Branch out from a given vertex.

Start at a vertex v ← chosen arbitrarily

add v to our subgraph H which we are building.

Each step, add an edge and incident vertex of minimal weight such that exactly one of its vertices (ends) is in H already.

Continue until all remaining edges have both ends in H.

(H built so its always connected)

maximal acyclic ⟹ tree.

minimal weight ?

if not minimal, choose a minimal one (H') who shares first k edges w/ H $e_1, \dots, e_k$ (in order of construction of H) w/ k maximal.

$H' + e_{k+1}$ has a cycle since H' maximal acyclic

cycle not in H

$e_{k+1} \in H$ so $\exists e_0$ in cycle, in H' not in H

in construction $e_0, e_1, \ldots, e_k$ is still acyclic (since in H' acyclic)

$w(e_0) \geq w(e_{k+1})$

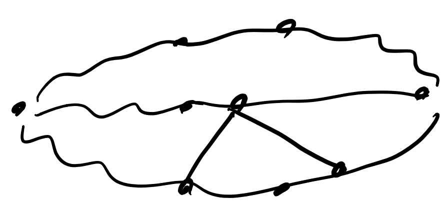
now $H' + e_{k+1} - e_0$ is at least as light as H' and has one more edge in common.

Intro to next topic: Connectivity & cuts

Two competing concepts of connectivity

- robustness / fragility – how many edges need to be removed to disconnect graph?

- redundancy – how many distinct ways can you go between different entities?



Def $\kappa'(G)$ = min'l # of edges whose removal makes G disconnected (or trivial)

$\kappa(G)$ = min'l # of vertices whose removal makes G disconnected (or trivial)

Def (today) G is *trivial* if it has 0 or 1 vertex.

Theorem $\kappa(G) \leq \kappa'(G) \leq \delta(G)$

$\delta(G)$ = minimal degree of a vertex.