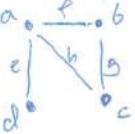


① graph theory

All graphs in this lecture are finite.

Def: A simple graph $G = (V, E)$ has vertices V and edges $E \subseteq \{\{u, v\} \mid u, v \in V\}$.

E.g.: 
 $f = \{a, b\}, e = \{a, d\}, \dots$

We say vertices a and b are adjacent if there is an edge f ,

$$f = \{a, b\} \in E, a \neq b$$

A vertex a is incident to an edge f if $a \in f$.

A graph allows loops which is an edge of the form $\{a\}$, a multigraph is a graph where two vertices may be connected by several edges (E is now a multiset formally) and a vertex may have several loops

E.g: Loop:



multigraph:

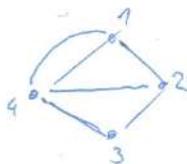


A weighted graph is a (multi)graph with a weight-function $w: E \rightarrow \mathbb{R}(I)$.

$N(u) = \{v \in E \mid \exists e \in E: e = \{u, v\}\}$ is the set of neighbors of a vertex u .

$d(u) = |N(u)|$ is the degree of a vertex u in a simple graph. $|outgoing\ edges| = d(u)$ in a multigraph. A graph is regular of degree r if $\forall u \in V: d(u) = r$.

E.g.:



$$N(1) = \{4, 2\},$$

$$d(1) = 3$$

Def: A directed (multi)graph is a graph where every edge has a head and a tail. Alternatively: Edges are pairs of vertices.

$$d^+(u) := |\{e \in E \mid e = (u, v) \text{ for some } v \in V\}| \quad \text{outdegree}$$

$$d^-(u) := |\{e \in E \mid e = (v, u) \text{ for some } v \in V\}| \quad \text{indegree}$$

Remark: A (di)graph can be regarded as a relation $u R v \iff (u, v) \in E$. R is symmetric if the graph is undirected.

Handshaking Lemma:

$$\sum_{v \in V} d(v) = 2|E|$$

(Here we use that a loop contributes 2 to the sum of the degree.)

Corollary: The number of vertices with an odd degree is even.

$$|\{v \in V \mid d(v) \text{ odd}\}| \in 2\mathbb{N}$$

E.g.: $K_n = (\{1, \dots, n\}, \{\{i,j\} \mid i \neq j\})$ is the complete graph on n vertices.



P_n is a path with n vertices.



C_n is a circle with n vertices.

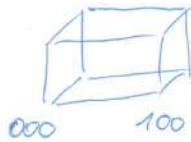


A hypercube: $V = \{0, 1\}^n$ (2^n vertices)

$$E = \{(u, v) \mid \sum_{i=1}^n |u_i - v_i| = 1\}$$

e.g.: $n=2$: $\begin{array}{c} 00 - 01 \\ | \quad | \\ 10 - 11 \end{array}$

$n=3$:



Def: The adjacency matrix $A = (a_{ij})$ of a graph is a $|V| \times |V|$ matrix with

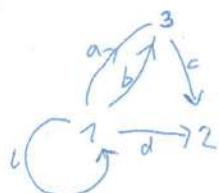
$$a_{ij} = \begin{cases} 1 & \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases} \quad \text{if } G \text{ is simple}$$

$$= \# \text{ edges } \{i, j\} \quad \text{if } G \text{ is a multigraph}$$

$$= \# \text{ edges } (i, j) \quad \text{if } G \text{ is a digraph}$$

$$= \sum_{(i, j) \in E} w(\{i, j\}) \quad \text{if } G \text{ is weighted and directed}$$

Eg:



$$\begin{pmatrix} 1 & b & a+b \\ 0 & 0 & 0 \\ 0 & c & 0 \end{pmatrix}$$

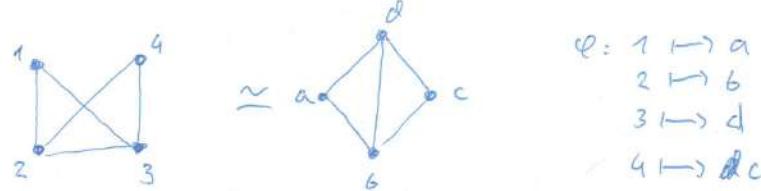
Warning: different graphs can have the same adjacency matrix because the labels are forgotten.

E.g.:

$$\begin{array}{c} \textcircled{a} \xrightarrow{\quad e \quad} \textcircled{b} \xrightarrow{\quad f \quad} \textcircled{c} \\ \textcircled{1} \quad \textcircled{2} \quad \textcircled{3} \end{array} \quad \left(\begin{array}{ccc} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array} \right) \text{ or } \left(\begin{array}{ccc} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{array} \right)$$

Def: Two simple graphs G, H are isomorphic $G \cong H$ if there is a bijective function $\varphi: V(G) \rightarrow E(H)$ such that $\{u, v\} \in E(G) \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E(H)$.

E.g.:



$$\begin{aligned} \varphi: 1 &\mapsto a \\ 2 &\mapsto b \\ 3 &\mapsto c \\ 4 &\mapsto d \end{aligned}$$

The drawing of two graphs doesn't make it clear whether they are isomorphic. It is unknown (tl today) whether there is an algorithm that decides in polynomial time if two graphs are isomorphic. (actual bound: quasi-polynomial)

Def: A walk/trail/path is a sequence $u_1, e_1, u_2, e_2, \dots$ of vertices u_1, \dots, u_k and edges e_1, \dots, e_{k-1} such that $e_i = \{u_i, u_{i+1}\}$.

A trail has no repeated edges, a path no repeated vertices.

A closed walk/circuit/cycle is a walk/trail/path with $e_k = \{u_k, u_1\}$.

The length of a closed walk is the number of edges.

Lemma: Let A be the adjacency matrix of a (multi)graph. Then $(A^k)_{i,j}$ is the number of walks from i to j of length k .

Proof: (induction), $k=0$: $A^0 = I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$ and the number of walks from i to j of length 0 equals to δ_{ij} .

$$k \mapsto k+1: (A^{k+1})_{i,j} = (A \cdot A^k)_{i,j} \stackrel{\substack{\text{row} \cdot \text{column} \\ \downarrow}}{=} \left(\sum_{v \in V} A_{iv} (A^k)_{vj} \right)_{i,j}$$

A walk from i to j of length $k+1$ is a walk from i to v of length one followed by a walk of length k from v to j . □

Def: A graph is (strongly) connected if for any two vertices u, v there is a walk from u to v . A digraph is weakly connected if the underlying graph (without direction) is connected. A bridge in a graph is an edge whose removal increases the number of connected components.

Def: H is a subgraph of G if H is a graph (of the same type) with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$

Def: A (simple) graph is bipartite if its vertices can be coloured with two colors red and blue such that edges only connect vertices of different colours.

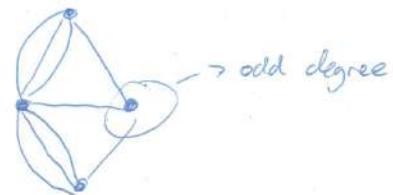
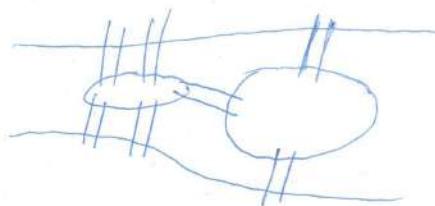
Theorem (König 1936): G is bipartite if and only if G contains no cycle of odd length.

Proof: \Rightarrow : Every cycle visits the same number of blue and red vertices.

\Leftarrow : w.l.o.g. G is connected. Fix u and color it blue. For any path from u to v of odd length color v red and for even length color v blue. This works cause of even cycle length. \square

Def: A Eulerian trail is a trail that uses every edge exactly once.

Theorem A connected has a Eulerian circuit if and only if all its vertices have an even degree.



Proof: \Rightarrow In any circuit every vertex is entered as often as it is used as a point of departure.

\Leftarrow : Induction on the number of edges: 0 edges $\Rightarrow G = \emptyset$ It is connected, has only one vertex. (A graph without vertices and no edges is disconnected.)

Otherwise Let W be any circuit in G (this exists, start anywhere, chose any edge unused so far, continue until you hit a starting vertex) Why is there an edge you can use? \rightarrow even degree condition.

Let $G' = (V(G), E(G) \setminus E(W))$, G' need not be connected, let G_1, \dots, G_n be the connected components of G' . In each component we can find an Eulerian circuit by induction since they have strictly fewer edges than G . W and G' have at least one vertex in common. Therefore W_1, \dots, W_n, W can be continued to an Eulerian circuit. \square

Trees and forests

Def: A forest is a graph without cycles (aka acyclic). A leaf is a vertex of degree 1.
A tree is a connected forest.

Lemma: If T is a tree on at least 2 vertices, it has at least 2 leafs.

Proof: $V(T)$ and $E(T)$ are finite. $\Rightarrow T$ contains a maximal path which has 2 leafs. \square

How's clear it has to be a maximal path? If one of the two vertices is connected to another vertex it would ~~not~~ be finite.

Definition: A subgraph H of a graph G is spanning if the vertices are the same.

Theorem B: A connected graph has a spanning tree.

We can remove edges as long as necessary and the remainder is a tree.

Theorem A: Let T be a graph. Then the following are equivalent:

- (1) T is a tree
- (2) Any two vertices are connected by a unique path
- (3) T is connected and every edge is a bridge (minimally connected). If you remove just one edge it is disconnected.
- (4) T has no cycles and adding any edge yields a cycle (maximally acyclic)

Theorem C: A graph is a tree if and only if it is connected and $|V| = |E| + 1$.

Proof A. (1) \Rightarrow (2): Otherwise it would not be connected or would have a cycle.

(2) \Rightarrow (3): Since any two vertices are connected, T is connected. Suppose $e \in E$ is no bridge $\Rightarrow E \setminus \{e\}$ would either leave a disconnected graph or ~~the~~ the path would not have been unique.

(3) \Rightarrow (4): An edge in a cycle can't be a bridge. Since T is connected adding an edge would create a cycle.

(4) \Rightarrow (1) If adding an edge yields a cycle $\Rightarrow T$ is connected. since a cycle would give 2 paths between any two vertices in the cycle. \square

Proof of B: As long as there is a non-bridge, remove it (using (3) of theorem A). \square

Proof of C: Induction on $|V|$: $|V|=1 \Rightarrow |E|=0$, $|V|=2 \Rightarrow 2=1+1$.

Let T' be T with a leaf removed, $|V(T')| = |V(T)| - 1$, $|E(T')| = |E(T)| - 1$

$$\Rightarrow |V(T)| = |V(T')| + 1 + |E(T')| + 1 + 1 = |E(T)| + 1$$

Let T' be a spanning tree of T

$$|V(T)| = |E(T)| + 1, |V(T')| = |E(T')| + 1$$

$$|V(T)| = |V(T')| \Rightarrow |E(T)| = |E(T')| \Rightarrow T = T'$$

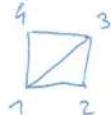
□

Matrix-tree theorem

How many spanning trees are in a graph?

~~Matrix-tree theorem~~

E.g.:



has 8 spanning trees: $\leftarrow \sqcup \sqcap \sqcap \sqcap \sqcap \sqcap \sqcap \sqcap$

theorems: (deletion - contraction)

~~Matrix-tree theorem: The following gives a way to obtain the number of spanning trees.~~

let $\tau(G) := \# \text{ spanning trees}$

$G \setminus e$ graph obtained by removing e

G/e graph obtained by contracting e

$$\Rightarrow \tau(G) = \tau(G \setminus e) + \tau(G/e)$$

E.g.: $\tau\left(\begin{array}{|c|c|} \hline 4 & 3 \\ \hline \boxed{2} & 1 \\ \hline 1 & 2 \\ \hline \end{array}\right) = \tau\left(\begin{array}{|c|c|} \hline 4 & 3 \\ \hline \quad & 1 \\ \hline 1 & 2 \\ \hline \end{array}\right) + \tau\left(\begin{array}{|c|c|} \hline 4 & 3 \\ \hline \quad & 1 \\ \hline \quad & 2 \\ \hline \end{array}\right) = 4 + 4$

Proof: The set of spanning trees is the disjoint union of spanning trees containing e and spanning trees not containing e .

More general: If G is a weighted graph with $w: E(G) \rightarrow \mathbb{R}$

Def Let H be a subgraph of G . Then $w(H) := \prod_{e \in E(H)} w(e)$

$\Upsilon(G)$ is the sum of the weights of the spanning trees of G , $\sum_{\text{spanning tree } H \subseteq E(G)} w(H) = \Upsilon(G)$

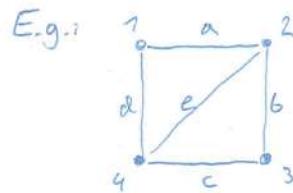
E.g.: $\tau\left(\begin{array}{|c|c|c|} \hline a & & c \\ \hline b & \boxed{e} & d \\ \hline & c & b \\ \hline \end{array}\right) = \tau\left(\begin{array}{|c|c|c|} \hline a & & c \\ \hline b & \quad & d \\ \hline & c & b \\ \hline \end{array}\right) + e \cdot \tau\left(\begin{array}{|c|c|c|} \hline a & & c \\ \hline b & \quad & d \\ \hline & c & b \\ \hline \end{array}\right) = abc + abd + acd + bcd + e(a+d)(b+c)$

Def The degree matrix of a graph is $\Delta = \text{diag}(d(v_1), \dots, d(v_n))$.

The degree in a weighted graph is $d(v) = \sum_{u \sim v, u \in E(G)} w(u, v)$.

Theorem: Let $n = |V(G)|$, $\lambda_1, \dots, \lambda_n$ the eigenvalues of $D-A$, one of these equals 0, w.l.o.g.: $\lambda_n = 0$. Then $\tilde{\chi}(G) = \frac{1}{n} \lambda_1 \cdots \lambda_{n-1}$

Equivalently: $\tilde{\chi}(G) = \det((D-A)_{i,i})$ where $(D-A)_{i,i}$ is obtained by removing row i and column i



$$\begin{matrix} & 1 & 2 & 3 & 4 \\ 1 & a+d & a & 0 & -d \\ 2 & a & a+b+c & -b & -c \\ 3 & 0 & -b & b+c & -c \\ 4 & -d & -c & -c & a+d \end{matrix} = D-A$$

$$\begin{aligned} \det((D-A)_{4,4}) &= (a+d) \begin{vmatrix} a+b+c & -b \\ -b & b+c \end{vmatrix} + (-a) \begin{vmatrix} a & 0 \\ -b & b+c \end{vmatrix} \\ &= (a+d)((a+b+c)(b+c) - b^2) - a^2(b+c) \end{aligned}$$

Spanning tree of minimal weight

Kruskal: G connected graph

sort edges by weight $w(e_1) \leq \dots \leq w(e_m)$

$$T_1 = \emptyset$$

For i from 1 to m :

if $E(T_i) \cup e_i$ is acyclic

$$T_{i+1} := T_i \cup e_i$$

else

$$T_{i+1} := T_i$$

if $|E(T_{i+1})| = n-1$

return

Remark: Kruskal is a "greedy algorithm": add best looking edge

Theorem: Kruskal yields a spanning tree of minimal weight

Proof: ~~By contradiction.~~

•) T is acyclic by construction

•) The algorithm reaches the return statement, suppose T_{m+1} is not connected

④ $e_i \rightarrow B$, $e_i \in G$, then Kruskal would have added e_i because $T_i \subseteq T_{m+1}$ is acyclic.

•) T is of minimal weight: see proof in matroid setting

□

Matroids provide a framework for greedy algorithms.

Def Let E be a set, I (the set of independent sets) a set of subsets of E

$$(M_1) \emptyset \in I$$

$$(M_2) B \in I, A \subseteq B \Rightarrow A \in I$$

$$(M_3) A, B \in I, |B| = |A| + 1 \Rightarrow \exists e \in B \setminus A : A \cup e \in I \quad ("exchange\ Axiom")$$

Then (E, I) is a matroid, if M_1 and M_2 are satisfied it's called an independent system.

Theorem G is a graph, $I := \{F \subseteq E(G) \mid F \text{ acyclic}\}$

$\Rightarrow (E, I)$ is a matroid

Proof: $M_1: \emptyset$ is a forest. $M_2: B$ is a forest $\Rightarrow A \in B$ is a forest.

$M_3: A, B$ edge sets of spanning forests, $|B| = |A| + 1$, find edge $e \in B \setminus A$ such that $A \cup e$ is a forest: Suppose A has connected components T_1, \dots, T_c

Show: $\exists e \in B \setminus A$ that is not in any of these components. count edges

$B|_{V(T_i)}$ is a forest $\Rightarrow |E(B|_{V(T_i)})| \leq |V(T_i)| - 1$

$$\Rightarrow \sum_{i=1}^c |E(B|_{V(T_i)})| \leq \sum_{i=1}^c (|V(T_i)| - 1) = |A|$$

B has more than $|A|$ edges so there exists an edge not in any $B|_{V(T_i)}$

□

Greedy: (E, I) matroid, $w: E \rightarrow \mathbb{R}$

sort E by weight $w(e_1) \leq \dots \leq w(e_m)$

$$T_i = \emptyset$$

for $i = 1 \dots m$

if $T_i \cup e_i \in I$

$$T_{i+1} := T_i \cup e_i$$

else

$$T_{i+1} := T_i$$

return $T := T_{m+1}$

Definition: A basis of a matroid is an inclusionwise maximal independent set.

Theorem: Greedy returns a basis of minimal weight, that is $\sum_{e \in T} w(e)$ is minimal among all bases.

Proof: T is a maximal independent set as in Kruskal.

$\sum w(e)$ minimal: let $T = \{t_1, \dots, t_s\}$, $w(t_1) \leq \dots \leq w(t_s)$. Suppose $B = \{b_1, \dots, b_n\}$, $w(b_1) \leq \dots \leq w(b_n)$

is a basis with $\sum_{b \in B} w(b) < \sum_{e \in T} w(e)$. Let $i := \min \{j \mid w(b_j) < w(t_j)\}$

Let $T_{i-1} = \{t_1, \dots, t_m\}$, $B_i = \{b_1, \dots, b_j\}$

$$\stackrel{M_3}{\Rightarrow} \exists b_j \in B_i \setminus T_{i-1} : T_{i-1} \cup b_j \in I$$

$w(b_j) \leq w(b_i) < w(t_i)$, so greedy should have chosen b_j instead of t_i .

j with $w(b_j) < w(t_i)$ exists because all bases have the same cardinality. (exercise)

□

Theorem: Suppose (E, I) satisfies M_1, M_2 and that for any weight function $w: E \rightarrow \mathbb{R}$ the greedy algorithm produces a maximal independent set $A \in I$ such that $\sum_{e \in A} w(e)$ is minimal along all maximal sets in I . Then (E, I) satisfies M_3 , hence is a matroid.

Proof: All maximal sets in I have the same cardinality. Suppose $A, B \in I$ maximal, $|A| < |B|$. Let $w(e) = \begin{cases} 1 & e \in B \\ 1 + \varepsilon & \text{otherwise} \end{cases}$ (we will determine $\varepsilon > 0$ in a suitable way)

for any $\varepsilon > 0$ greedy returns B

$$w(B) = |B| \geq |A| + 1$$

$$w(A) = |A \cap B| + (1 + \varepsilon)|A \setminus B| = |A| + \varepsilon|A \setminus B|$$

$$\Rightarrow \text{choose } \varepsilon < \frac{1}{|A \setminus B|}$$

Show (E, I) satisfies M_3 : Let $A, B \in I$, $|B| = |A| + 1$, $w(e) = \begin{cases} 0 & e \in A \\ 1 & e \in B \setminus A \\ x & \text{otherwise} \end{cases}$ (to be terminated)

greedy chooses all of A first, since $|A| < |B|$ A is not maximal.

Suppose $\# e \in B \setminus A = r$: $A \cup e \in I \Rightarrow$ greedy chooses $r - |A|$ elements of weight x , where r is the size of any basis. Call this set A'

$$\text{Also: } \exists \text{ basis } B' = B \cup \{e_1, \dots, e_{r-|A|}\}, w(A') = \underbrace{w(A)}_0 + x(r - |A|) = x(r - |A|)$$

$$w(B') \leq |B \setminus A| + x(r - |B|) = |B \setminus A| + x(r - |A| - 1) \leq w(A') + |B \setminus A| - x$$

\Rightarrow choose $x > |B \setminus A|$, then $w(B') < w(A)$, but greedy returned A' □

□

Prims algorithm: G connected, r any vertex

$$Q := V(G) \setminus r, T := \emptyset, V_T := \{r\}$$

while $Q \neq \emptyset$

$v :=$ a vertex in Q connected to T with an edge e of minimal weight

$$Q = Q \setminus v$$

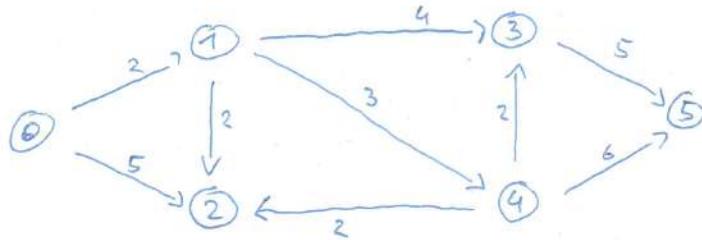
$$T = T \cup \{e\}, V_T = V_T \cup \{v\}$$

Prims is a greedy algorithm, but there is no matroid underlying.

Networks

Take G , a weighted directed graph. The distance (length) of a path P is $\sum_{e \in P} w(e)$.

E.g.



What's the best path from 0 to 5 ? We present 3 algorithms:

1.) Dijkstra's (1950), single source, only for $w: E \rightarrow \mathbb{R}^+$

Time complexity: $\mathcal{O}(V \log V + |E|)$

2.) Bellman-Ford-Moore, single source, $w: E \rightarrow \mathbb{R}$ (allows negative weights)

Time complexity: $\mathcal{O}(V|E|)$

3.) Floyd-Warshall, for all distances, works in $\mathcal{O}(V^3)$ time

Dijkstra's algorithm

$$d(v) = \begin{cases} 0 & v = v_0 \\ \infty & \text{otherwise} \end{cases}$$

$$Q = V$$

while $Q \neq \emptyset$

Find $u \in Q$ with $\min d(u)$

$$Q = Q \setminus \{u\}$$

for $v \in Q$, $(u, v) \in E$

$$d(v) = \min(d(v), d(u) + w(u, v))$$

If $d(u) + w(u, v) < d(v)$

$$d(v) = d(u) + w(u, v)$$

E.g.: We use Dijkstra's algorithm on the graph above.

$d(v)$	0	1	2	3	4	5
0	∞	∞	∞	∞	∞	∞
0	2	5	∞	∞	∞	∞
0	2	4	6	5	7	∞
0	2	4	6	5	10	∞

Remark: Dijkstra's algorithm chooses the min weight edge between Q and $V \setminus Q$. This is called breadth-first-search.

Bellman - Ford - Moore Algorithm

$$d(v) = \begin{cases} 0 & v = v_0 \\ \infty & \text{otherwise} \end{cases}$$

$$L(v) = \text{length of path} = \begin{cases} 0 & v = v_0 \\ \infty & \text{otherwise} \end{cases}$$

step = 0

while True

modified = false

for $u \in V$

$$L(u) = \text{step}$$

for $e = (u, v) \in E$

$$\text{if } d(v) > d(u) + w(e)$$

modified = True

$$d(v) = d(u) + w(e)$$

$$L(v) = L(v) + 1$$

if not modified

return d

else if step = $|V| - 1$

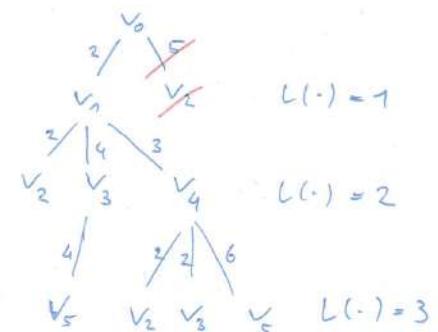
return negative cycle (error)

else

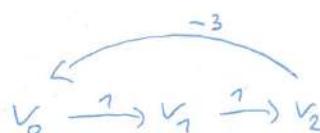
$$\text{step} = \text{step} + 1$$

Example:

$$\Rightarrow v_0, L(v_0) = 0, d(v_0) = 0$$



We give an example of the execution of the BFM algorithm on a graph with a negative cycle:



$$v_0 \xrightarrow{1} v_1 \xrightarrow{1} v_2 \xrightarrow{-3} v_0, \text{ step} = 2$$

Floyd - Warshall - Algorithm

$$d(u, v) = \begin{cases} 0 & u = v \\ w(u, v) & (u, v) \in E \\ \infty & \text{otherwise} \end{cases}$$

$\rightarrow \approx$ adjacency matrix

For $u \in V$

for $v \in V$

for $w \in V$

$$d^{(v, w)}_{\min} = \min\{d^{(v, w)}, d^{(v, u)} + d^{(u, w)}\}$$

If $d(v, v) < 0$

error negative cycle

Flows

Let G be a weighted graph with a non-negative weight function $w: E \rightarrow \mathbb{R}^+$. Take a node s as the source in V ($d^-(s) = 0$) and a node t as the target/sink in V ($d^+(t) = 0$). Then $\phi: E \rightarrow \mathbb{R}$ is a flow if

$$1.) \forall e \in E : 0 \leq \phi(e) \leq w(e)$$

$$2.) \forall u \in V \setminus \{s, t\} : \sum_{\substack{v \\ (v,u) \in E}} \phi(v, u) = \sum_{\substack{u \\ (u,v) \in E}} \phi(u, v)$$

The value of the flow ϕ is $\text{val}(\phi) = \sum_{\substack{u \\ (s,u) \in E}} \phi(s, u)$.

Lemma:

We will prove that $\text{val}(\phi) = \sum_{\substack{u \\ (u,t) \in E}} \phi(u, t)$

$$\begin{aligned} \text{Proof: } \sum_{\substack{u \\ (s,u) \in E}} \phi(s, u) + \underbrace{\sum_{\substack{v \neq t \\ (v,u) \in E}} \sum_{\substack{u \\ (v,u) \in E}} \phi(v, u)}_{=} &= \sum_{e \in E} \phi(e) \quad \text{and} \\ \sum_{\substack{u \\ (u,t) \in E}} \phi(u, t) + \underbrace{\sum_{\substack{v \neq t \\ (u,v) \in E}} \sum_{\substack{u \\ (u,v) \in E}} \phi(u, v)}_{=} &= \sum_{e \in E} \phi(e) \end{aligned}$$

□

Def: Let $S \subseteq V$, $s \in S$, $t \notin S$. Then $(S, V \setminus S)$ is called a cut. Any edge from S to $V \setminus S$ is called crossing the cut. The capacity of a cut is $c(S, V \setminus S) = \sum_{\substack{(u,v) \in E \\ u \in S, v \notin S}} w(u, v)$.

A cut is minimal if its capacity is minimal among all cuts, a flow is maximal if its value is maximal among all flows.

Lemma $\text{val}(\phi) \leq c(S, V \setminus S)$ for any ϕ and all cuts $(S, V \setminus S)$

Proof:

$$\begin{aligned} \text{val}(\phi) &= \sum_{u \in S} \left(\underbrace{\sum_{\substack{v \\ (v,u) \in E}} \phi(v, u)}_{=0 \text{ except for } u=s} - \underbrace{\sum_{\substack{v \\ (u,v) \in E}} \phi(u, v)}_{\substack{\uparrow \text{edges from } S \text{ to } V \setminus S \\ \uparrow \text{edges from } V \setminus S \text{ to } S}} \right) = \sum_{u \in S} \phi(u, v) - \sum_{v \in V \setminus S} \phi(u, v) \leq c(S, V \setminus S) \end{aligned}$$

□

Def: An augmenting path P for a flow ϕ is an (unoriented) path from s to t with
 $\forall e \in P$ traversed in forward direction: $\phi(e) < w(e)$
 $\forall e \in P$ traversed backwards: $\phi(e) > 0$

Theorem: Let ϕ be a flow. Then the following are equivalent:

- (1) $\text{val}(\phi)$ is maximal
- (2) there is no augmenting path for ϕ
- (3) $\text{val}(\phi) = c(S, V \setminus S)$ for some S

Proof: (1) \Rightarrow (2). Suppose P is an augmenting path. Construct a bigger flow:

$$\delta_1 := \min_{\substack{e \in P \\ \text{forward}}} w(e) - \phi(e), \quad \delta_2 := \min_{\substack{e \in P \\ \text{backward}}} \phi(e), \quad \varepsilon := \min \{\delta_1, \delta_2\}$$

$$\tilde{\phi}(e) := \begin{cases} \phi(e) + \varepsilon & e \text{ forward edge} \\ \phi(e) - \varepsilon & e \text{ backward edge} \\ \phi(e) & \text{otherwise} \end{cases} \text{ is a bigger flow}$$

(2) \Rightarrow (3) Suppose there is no augmenting path, construct the cut

$$S = \{v \in V \mid \exists \text{ augmenting path from } s \text{ to } v\}$$

$s \in S, t \notin S \Rightarrow (S, V \setminus S)$ is a cut. For forward crossing edges we have $\phi(e) = w(e)$, for backwards crossing edges $\phi(e) = 0$ (otherwise there would be an augmenting path.) The previous lemma then gives (from the proof of it)

$$\text{val}(\phi) = \sum_{\text{forward}} \phi(e) - \sum_{\text{backward}} \phi(e) = c(S, V \setminus S) - 0$$

(3) \Rightarrow (1) Start with a cut such that $\text{val}(\phi) = c(S, V \setminus S)$ which already gives that the flow is maximal since for any S $\text{val}(\phi) \leq c(S, V \setminus S)$

□

Theorem: A maximal flow exists

Proof: If $w: E \rightarrow \mathbb{N}$ an augmenting path increases $\text{val}(\phi)$ by at least one. For rational weights multiply. If $w: E \rightarrow \mathbb{R} =$ Any continuous function (like $\text{val}: \{\phi\} \rightarrow \mathbb{R}$) on a compact set has a maximum.

□

Corollary val of max-flow = capacity of min-cut.

Ford Fulkerson Algorithm

$$\phi_i(e) := 0 \quad \forall e \in E$$

while \exists augmenting path for ϕ :

$$\phi_{i+1}(e) = \begin{cases} \phi_i(e) + \varepsilon & e \text{ forward} \\ \phi_i(e) - \varepsilon & e \text{ backward} \\ \phi_i(e) & \text{otherwise} \end{cases}$$

Def $M \subseteq D$ for $D \subseteq X \times Y$ is called a perfect matching if

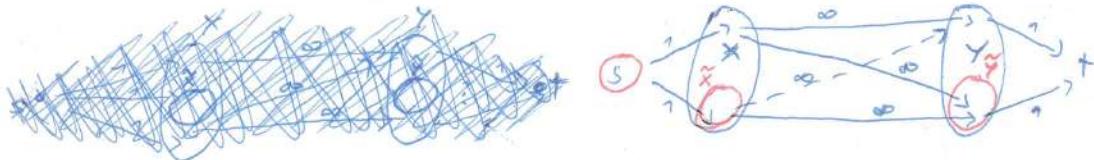
$$\forall x \in X \quad \exists! y \in Y : (x, y) \in M.$$

Hall's marriage theorem X, Y finite, disjoint sets, $|X| = |Y|$, $D \subseteq X \times Y$

D admits a perfect matching $\Leftrightarrow \forall w \in X : |N(w)| \geq |w|$

Proof: \Rightarrow : use perfect matching.

\Leftarrow : add a source with weighted edge of one to X , put ∞ (or a large number) on edges from X to Y , edges with weight one from Y to a target t . Then the value of a max-flow is the size of the largest matching.



$(\{s\} \cup \tilde{X} \cup \tilde{Y}, X \setminus \tilde{X} \cup Y \setminus \tilde{Y} \cup \{t\})$ is a minimal cut, \tilde{X}, \tilde{Y} being the matching

- $\tilde{Y} \supseteq N(\tilde{X})$ because $w(x, y) = \infty$
 - $c(S, V \setminus S)$, $S := \{s\} \cup \tilde{X} \cup \tilde{Y}$, $c(S, V \setminus S) = |X \setminus \tilde{X}| + |\tilde{Y}|$
 - $c(S, V \setminus S) \geq |X|$ because $|\tilde{Y}| \geq |N(\tilde{X})| \geq |\tilde{X}|$
↑
hypothesis
- $\Rightarrow |X \setminus \tilde{X}| + |\tilde{Y}| \geq |X \setminus \tilde{X}| + |\tilde{X}| = |X|$ □

Hamiltonian Graphs (simpler undirected, unweighted)

Def A graph G is hamiltonian if it contains a hamiltonian cycle, which visits every vertex exactly once.

Remark: It is NP-hard to find a hamiltonian cycle. (to check if G is hamiltonian)
RL good approach

Def Let G be a graph, let $[G] = (V, \tilde{E})$ with ~~all edges~~

$$E_1 := E, E_{i+1} := \{E_i \cup \{e = (u, v) \notin E_i \mid d(u) + d(v) \geq |V|\}\}$$

\tilde{E} is the set E_k such that $d(u) + d(v) < |V|$ for all $(u, v) \notin E_k$
is called closure of G .

Theorem G is hamiltonian $\Leftrightarrow [G]$ is hamiltonian

Proof: \Rightarrow trivial (there are only edges added to the closure)

\Leftarrow : suppose H is a hamiltonian cycle in the closure, but G is not hamiltonian

~~$\exists e = (u, v) \in E([G]) \setminus E(G)$: consider step by step adding edges to H to consider steps in the closure which creates the cycle~~

~~Label the edges of H : $h = h_1, \dots, h_n = h$~~

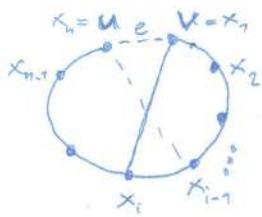
~~Correct proof in next lecture~~

~~$\forall i \in \{2, \dots, n-1\} : h_i \in E(G) \Rightarrow (u_i, v_i) \in E(G)$ for $i \in \{2, \dots, n-1\}$~~

~~$\Rightarrow d_G(u) \leq \# \text{vertices other than } u \text{ not directly below a neighbour of } v$~~

~~overall shortest cycle $= n - 1 - d_G(v) \Rightarrow d_G(u) + d_G(v) \leq n - 1$~~

Let G_i be the step in creating the closure where the first edge $e = (u, v) \in E([G_i]) \setminus E(G)$ appears. Label the vertices in H : $v = x_1, \dots, x_n = u$



$$(v, x_i) \in E(G) \Rightarrow (v, x_{i+1}) \notin E(G) \text{ for } i \in \{2, \dots, n-1\}$$

$$\Rightarrow d_G(u) \leq \# \text{ vertices other than } u \text{ not directly before a neighbor of } v \\ = n-1 - d(v)$$

$$\Rightarrow d_G(u) + d_G(v) \leq n-1$$

$\Rightarrow (u, v)$ would not have been put into G_i in constructing $[G]$ \square

Corollary: If $|V| \geq 3$, $d(u) + d(v) \geq |V| \forall u, v \Rightarrow G$ is hamiltonian (Ore 1960)

Corollary: $d(v) \geq \frac{|V|}{2} \forall v \in V \Rightarrow G$ is hamiltonian (Dirac 1952)

Planarity

Def A graph is planar if there is a drawing of G in \mathbb{R}^2 such that no two edges intersect (except at vertices).

E.g.  is planar since it can be drawn like this: 

Theorem: $K_{3,3}$ and K_5 are not planar.



Def A face of a drawing of a graph is a region bounded by edges.

Theorem (Euler's polyhedron formula): $|V| - |E| + |F| = 2$ for any drawing of a planar graph G , G connected.

Proof: induction on $|F|$: $|F| = 1 \Rightarrow G$ is a tree $\Rightarrow |V| - |E| + |F| = 2$

$|F| \geq 2$: $\exists e$ bounding two faces, let $G' = G \setminus e$, (e is part of a cycle cause it bounds 2 faces)

$$\Rightarrow |V'| = |V|, |E'| = |E| - 1, |F'| = |F| - 1$$

$$\Rightarrow 2 = |V'| - |E'| + |F'| = |V| - (|E| - 1) + |F| - 1 = |V| - |E| + |F|$$

\square

Lemma G simpler planar, connected $\Rightarrow |E| \leq 3|V| - 6$.

G simpler planar, connected, no triangles $\Rightarrow |E| \leq 2|V| - 4$.

Condition: ~~connected~~ $|V| \geq 3$

Proof: $f_j := |\{f \text{ face} \mid f \text{ has } j \text{ bounding edges}\}| \Rightarrow |F| = \sum_{j \geq 3} f_j$ (≥ 3 cause simplicity)

$$\Rightarrow 3|F| \leq \sum_{j \geq 3} j f_j \leq 2|E| , \quad 0 = 3|V| - 3|E| + 3|F| - 6 \\ \leq 3|V| - 3|E| + 2|E| - 6 \\ = 3|V| - |E| - 6$$

no triangles: $4|F| \leq 2|E| : \quad 0 = 2|V| - 2|E| + 2|F| - 4 \\ \leq 2|V| - 2|E| + |E| - 4 = 2|V| - |E| - 4$

□

Corollary: $K_{3,3}$ is not planar, K_5 is not planar

Proof: $K_{3,3}$ has no triangles, $|V|=6$, $|E|=9$. K_5 : $|V|=5$, $|E|=\binom{5}{2}=10$

□

Theorem: (Kuratowski - Wagner): G is planar $\Leftrightarrow G$ has no subgraph which is a subdivision of $K_{3,3}$ or K_5 .

Graph colourings

Def: G undirected, simple, then a function $c: V \rightarrow \{1, \dots, r\}$ is called a proper vertex colouring if adjacent vertices have different colours. ($\{1, \dots, r\}$ are the colours). $\tilde{c}: E \rightarrow \{1, \dots, r\}$ is a proper edge colouring if incident edges have different colours.

Lemma: $G = (V, E)$, $\bar{G} = (\bar{V}, \bar{E})$ where $\bar{V} = E$, $(e, f) \in \bar{E} \Leftrightarrow e, f$ incident edges in G

then \tilde{c} is a proper edge colouring of $G \Leftrightarrow \tilde{c}$ is a proper vertex colouring in \bar{G}

Remark: \bar{G} is called the line graph of G

E.g.: G  \bar{G} 

Remark:  is not a line graph of any graph.

Def: The chromatic number $\chi(G)$ is the minimal number of colours in a proper vertex colouring of G .

E.g.: 1) $\chi(K_n) = n$,

2) G bipartite $\Rightarrow \chi(G) = 2$

Theorem: G planar $\Rightarrow \chi(G) \leq 4$ (4-colour theorem)

Proof: Appel & Haken (1976 ??) & Computer (reduced problem, computer checked remaining cases)

Theorem: G planar $\Rightarrow \chi(G) \leq 5$

Proof: If $d_{\min}(G) \leq 5$, because suppose $d_{\min}(G) \geq 6$

$$\Rightarrow 2|E| = \sum_{v \in V} d(v) \geq 6|V|$$

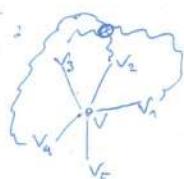
but we already had $|E| \leq 3|V| - 6$, \therefore that's a contradiction.

~~xxxxxxxxxxxxxx~~. Let v be a vertex of minimal degree and let $G' = G \setminus \{v\}$.

By induction $\chi(G') \leq 5$. ($\chi(\cdot) = 1$, use induction hypothesis on G')

case $d(v) \leq 5$: use ~~the~~ 5th colour on v

case $d(v) = 6$:



If v_1, \dots, v_5 use max 4 colours, use the ~~the~~ 5th for v .

Otherwise, wlog, assume v_i has colour i

Let $G'_{i,j}$ be the subgraph of G' whose vertices are coloured

with i or j , together with the corresponding edges.

If v_i and v_j are in different components of $G'_{i,j}$, switch the colours in one component.

$\Rightarrow v_1, \dots, v_5$ then have 4 colours.

Otherwise, suppose v_i, v_j are in the same component of $G'_{i,j}$ for all pairs i, j

$\Rightarrow \exists$ path P_1 : $v_i \rightsquigarrow v_3$ in $G'_{i,j}$

\exists path P_2 : $v_2 \rightsquigarrow v_4$ in $G'_{i,j}$

\Rightarrow there has to be a vertex in both paths. $\therefore (P_1, P_2)$ are intersecting

□

Def: The chromatic polynomial $\chi(G, x)$ is the number of proper vertex colourings of G with x colours

E.g. $\Rightarrow \chi(\Delta, x) = x(x-1)(x-2)$

$\Rightarrow \chi(P, x) = x(x-1)^3$

$\chi(\text{isolated}, x) = x^3$ complement, so n isolated vertices

$\chi(K_n, x) = x(x-1)\dots(x-n+1)$, $\chi(P_n, x) = x(x-1)^{n-1}$, $(K_n^c, x) = x^n$

Theorem: $\chi(G, x) = \chi(G \setminus e, x) - \chi(G/e, x)$ for any edge $e \in E$

Proof: any proper vertex colouring of $G \setminus e$ is either a proper vertex colouring of ~~of~~ G/e , $e = \{u, v\}$ (if the colours of u, v are the same) or a proper vertex colouring of G (if the colours of u, v are ~~not~~ distinct in $G \setminus e$). □

Theorem $\chi(G, x)$ is a polynomial.

Proof: Induction on $|E| + |V|$.

case $|E| = 0 \Rightarrow \chi(G, x) = x$ is a polynomial

case $|E| > 0$: $\chi(G, x) = \chi(G \setminus e, x) - \chi(G/e, x)$
 $\stackrel{!}{=} \text{polynomial} - \text{polynomial}$

□

② combinatorics

Given a finite set A_n for each $n \in \mathbb{N}$, what is $|A_n|^2$? e.g. $A_n = \{\pi \mid \pi \text{ permutation of } \{1, \dots, n\}\}$

An answer might be: o) an "explicit formula": $|A_n| = n!$

o) a recurrence: $|A_n| = n|A_{n-1}|$, $|A_0| = 1$

o) an asymptotic formula $|A_n| \sim n^n e^{-n} \sqrt{2\pi n}$, o) an algorithm

counting principles

Let A, B, C be finite sets

o) $A \cap B = \emptyset \Rightarrow |A \cup B| = |A| + |B|$

o) $|A \times B| = |A| \cdot |B|$

o) $f: |A| \rightarrow |B|$ bijective $\Rightarrow |A| = |B|$

Double counting: Let $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_m\}$, $R \subseteq A \times B$, $|R|^2$.

Let $R_{i,j} := \{(a_i, b_j) \in R\}$, $R_{*,j} = \{(a, b_j) \in R \mid a \in A\}$

$$\Rightarrow |R| = \sum_{j=1}^m R_{*,j} = \sum_{i=1}^n R_{i,*}$$

E.g., $\bar{c}(n) = \text{average number of divisors of integers between 1 and } n$

$$\bar{c}(6): \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \diagdown & \diagdown & \diagdown & \diagdown & \diagdown & \diagdown \\ 1 & 2 & 2 & 3 & 2 & 4 \end{matrix} \\ \text{1 divides everything,} \\ \text{2 divides even numbers,} \\ \vdots \quad \leftrightarrow \end{array} \Rightarrow \bar{c}(6) = (1+2+2+3+2+4) \frac{1}{6} = \frac{19}{6} = \frac{1}{6} (6+3+2+1+1+1) = \frac{1}{n} \sum_{i=1}^n \lfloor \frac{n}{i} \rfloor$$

↓
sum over rows
columns

sum over rows

$$\text{we only want to approximate: } \frac{1}{n} \sum_{i=1}^n \lfloor \frac{n}{i} \rfloor = \frac{1}{n} \sum_{i=1}^n \frac{n}{i} + O(1) = \sum_i \frac{1}{i} + O(1) \sim \log(n)$$

constant error

Pigeonhole - Principle (Schubfachprinzip)

A_1, \dots, A_k , pairwise disjoint, $|A_1 \cup \dots \cup A_k| > k \cdot r$

$$\Rightarrow \exists i: |A_i| > r$$

Alternatively: $f: |A| \rightarrow |B|$, $|A| > |B| \Rightarrow \exists b \in B: |f^{-1}(b)| \geq 2$ (not injective).