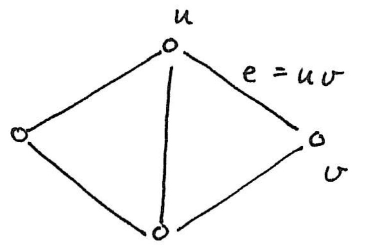
GRAPH THEORY

Graph: $G = (V, E)$ V: nodes, finite set
E: edges

edge: $e = [u, v] = uv$, unordered pair of nodes

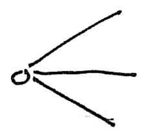
Ex



- u and v are <u>neighbours</u> if there is an edge between them (adjacent nodes)

- The <u>degree</u> of a node is the number of incident edges

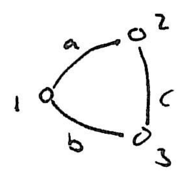
  Ex.

  $d = 3$

- Loops  and parallell edges  assumed not to occur

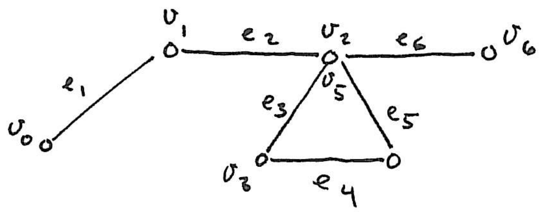
- Node-edge incidence matrix



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

- **Walk**: node-edge sequence $v_0, e_1, v_1, e_2, \ldots, e_m, v_m$ where e_i is an edge between v_{i-1} and v_i

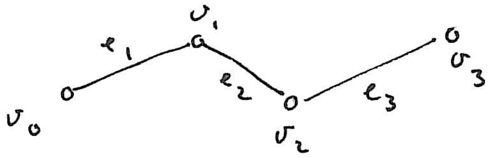
Ex



- **Path**: a walk with no repetition of nodes

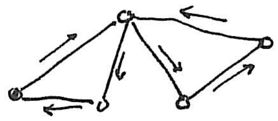
$P = v_0, e_1, v_1, \ldots, e_m, v_m$

ex

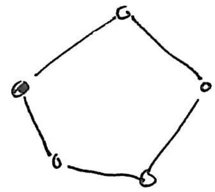


- If $s = v_0$, and $t = v_m$, P is called an st-path

- If $v_0 = v_m$, ~~the~~ walk is closed



- A closed walk where all nodes have degree 2 is called a cycle

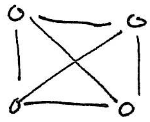


- The length of a walk is equal to the number of edges

- Some special graphs

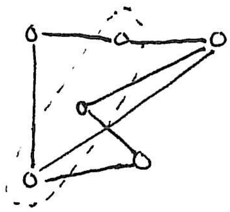
  - Complete graph, K_n

    ex. K_4



  - Bipartite graph: $V = V_1 \cup V_2$ with all edges going between V_1 and V_2

    ex.

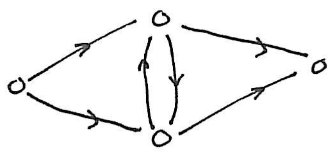


- <u>Directed graphs</u>

  $D = (V, A)$, V: nodes, <u>vertices</u>
  A: edges, <u>arcs</u>

  $a = (u, v)$; ordered pair of vertices

  Ex.

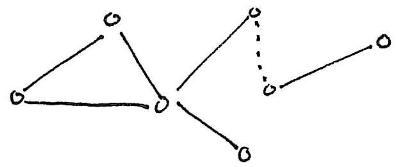


u: tail
v: head

  - Directed walk, path and cycle follows naturally

- A graph is <u>connected</u> if there is a path between all pair of nodes

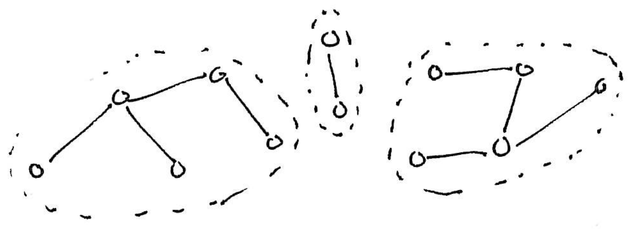
ex.



- A graph is <u>acyclic</u> if it does not contain any cycle.

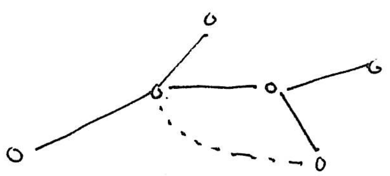
  An acylic graph is also called a <u>forest</u>.

- A <u>tree</u> is a connected acylic graph.
  - A tree is a connected forest
  - A forest is a union of trees.

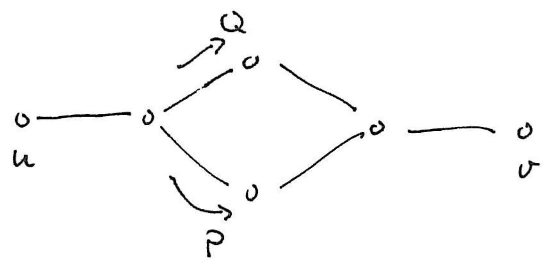


- <u>Proposition</u>

  $T = (V, E)$ is a tree $\Leftrightarrow$ there is a unique path between each pair of nodes.

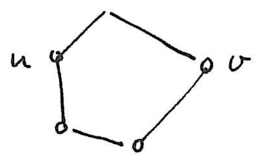


$\Rightarrow$ Let $u, v \in T$, $u \neq v$. Then there exists a uv-path P. Assume that there is another uv-path Q with $Q \neq P$.



Then $P \Delta Q$ contains a cycle contradicting the assumptions. Thus no such Q exists.

$\Leftarrow$ T is obviously connected.
Assume that T contains a cycle S.



Consider $u \neq v$ in S. Then S will give rise to two different paths between u and v, in contradiction with the assumptions.

$\square$

Exercises:

Prove Proposition 4.5 (p. 84)

⑥ MINIMUM SPANNING TREES

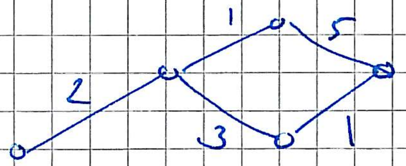
$G=(V,E)$: connected undirected graph

$\ell : E \to \mathbb{R}$, length function

For $F \subseteq E$, $\ell(F) := \sum_{e \in F} \ell(e)$

<u>Problem</u> : Find a spanning tree of minimum length

<u>Ex</u>



A forest F is <u>greedy</u> if there exists a min.-length spanning tree T of G that contains (the edges-set of) F.

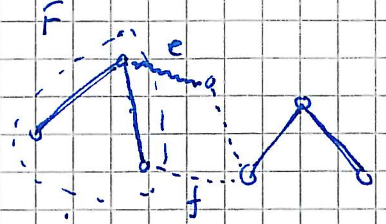
<u>Theorem 1.11</u>

F : greedy forest , U : one component , $e \in \delta(U)$.
If e ~~has minimum length~~ has minimum length among all edges in $\delta(U)$, then $F \cup \{e\}$ is again a greedy forest.

<u>Proof</u>

T : min. spanning tree containing F.
P : unique path in T between end vertices of e

(7)



φ: ~~orig~~ P contains ~~atl~~ at least one edge f that belongs to $\delta(U)$

$T' := (T \setminus \{f\}) \cup \{e\}$ also a tree

By assumption $l(e) \leq l(f)$ and $l(T') \leq l(T)$

T' is ~~an~~ min spanning tree

$F \cup \{e\} \subseteq T' \Rightarrow F \cup \{e\}$ is greedy

$\square$

~~Given~~ Two algorithms follow

- ~~Algorithm 1 (Dijkstra-Prim~~

Dijkstra-Prim algorithm (1959, 1957)

1. Choose vertex $v_1 \in V$ arbitrarily
   Let $U_1 = \{v_1\}$
2. Choose edge $e_{k+1} \in \delta(U_k)$ with min length
3. $U_{k+1} = U_1 \cup \{e_{k+1}\}$

. ~~Start~~ Choose a vertex
Build a tree by adding the vertex closest to a node in the tree

Running-time: $O(|V|^2)$

With heaps: $O(|E| \log |V|)$

Fibonacci heaps: $O(|E| + |V| \log |V|)$

(8) • Greedy algorithm - best choice, no need for backtracking

• Kruskal's algorithm

1. ~~Let $V_i = \{v_i\}$ for every $v_i \in V$~~ $T = \emptyset$

~~2.~~ ~~$S = \{e_1, \dots, e_m\}$~~ $S = E$

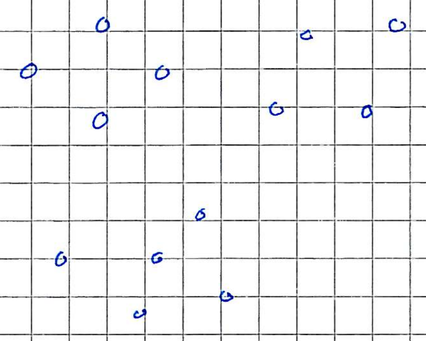
2. Find $e \in S$ of min. length such that $T \cup \{e\}$ is a forest

3. $T := T \cup \{e\}$, $S := S \setminus \{e\}$

Running-time: $O(|E| \log |E|)$

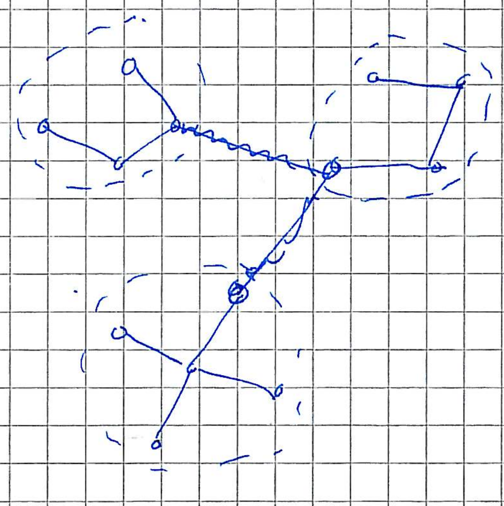
Build a forest by adding the ~~m~~ shortest edge remaining that do not create a cycle.

(9) <u>Ex</u> MST in Cluster analysis



1. Form a minimum spanning tree (based on some distance measure)

2. Remove edges according to some criteria e.g. K longest edges



Similar ideas used in image ~~anato~~ segmentation.