BIPARTITE GRAPHS, TOPOLOGICAL ORDERING

Marcus Ritt

CMP 601 – Algorithms and Theory of Computation — April 2020

Outline

1. Bipartite graphs

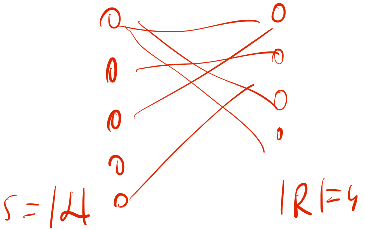
2. Topological sorting

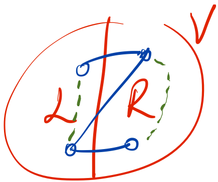
$L \subseteq V$
$R \subseteq V$

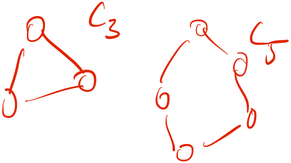
An undirected graph $G = (V, E)$ such that

- Vertices have two parts: $V = L \cup R$, $\boxed{L \cap R = \emptyset}$
- Edges only between parts: $E \subseteq \{\{u, v\} \mid u \in L, v \in R\}$

$V = L \dot\cup R$
Disjoint union

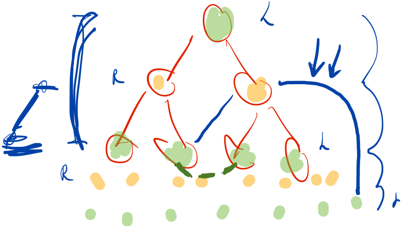






The following is equivalent:

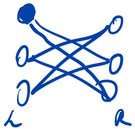
1. G is bipartite
2. G has no odd cycles
3. G has no edges connecting vertices in the same layer of the BFS tree.



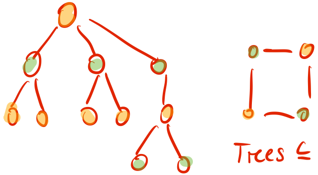
$\{u,v\} \in E$, $\{u,v\} \in T$

$\Rightarrow$ $\boxed{\text{Layer distance} \leq 1.}$

Proof idea

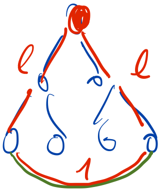


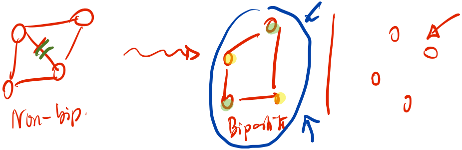
- $1 \rightarrow 2$: A cycle must end at the same vertex. This can be achieved only by an even number of edges, since the graph is bipartite.
- $2 \rightarrow 3$: An edge connecting vertices in the same layer would induce an odd cycle (with the least common ancestor)
- $3 \rightarrow 1$: Join the even layers and odd layers. Since no edges connect the same layer, the graph is bipartite.

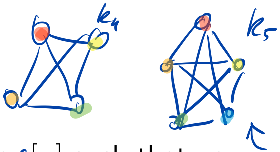


Trees ⊆ Bipartite

$2\ell+1$



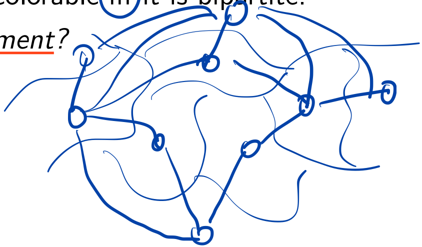




- Graph coloring: a map $c : V \to [n]$ to colors $[n]$ such that no adjacent vertices have the same color. *Goal*: minimize the number of colors. NP-complete

$K_n \to n$ colors

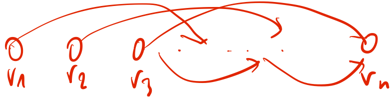
- Famously: planar graphs are 4-colorable.
- Observation: graph is 2-colorable iff it is bipartite.
- *Can you prove this statement?*



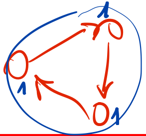
Definition

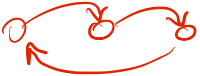
$n = |V|$ $m = |A|$

- We consider now directed graphs $G = (V, A)$.
- Topological order(ing): A vertex order $v_1, \ldots, v_n$ such that for all arcs $(v_i, v_j) \in A$ we have $i < j$.
- Algorithm that does it: topological sort



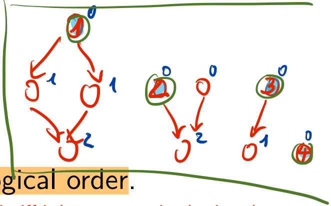
✓





X

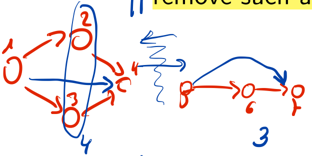
Directed acyclic graph



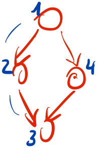
- Proposition: G is DAG iff G has a topological order.
- Proof idea:

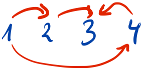
  For a directed graph G: G is acyclic iff it has a topological order.

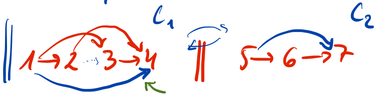
  - "$\leftarrow$": if $v_{i_1}, \ldots, v_{i_n}$ is a cycle, $i_1 < i_2 < \cdots < i_n < i_1$ a contradiction.
  - "$\rightarrow$": there must be a vertex $v \in V : \delta(v) = 0$. Repeatedly remove such a vertex.

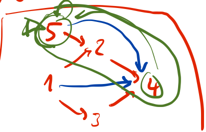


linear time:
$O(n+m)$



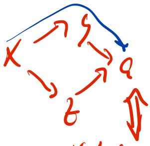






- Partial order $\preceq$ over set V
  1. Reflexive: $x \preceq x$
  2. Transitive: $x \preceq y, y \preceq z \rightarrow x \preceq z$
  3. Anti-symmetric: $x \preceq y, y \preceq x \rightarrow x = y$.

G



$x \leq y$ $\quad x \leq z$
$x \leq q$ $\quad y \leq q$
$\boxed{z \leq q}$

- DAGs represent partial orders:

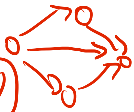
  $\boxed{x \preceq y} \Longleftrightarrow$ *there is a directed xy-path*

- Gives rise to *transitive closure* G^C of DAG G

  If $\boxed{x \preceq y}$ *add arc* (x, y) *to* G

  $x \leq y \Longleftrightarrow (x, y) \in A$

G^C



$O(1)$

- And: to *transitive reduction*

  Smallest graph G^R *that represents a given order* $\preceq$

G^R

- Exercise: find efficient algorithms for computing transitive closures and reductions.