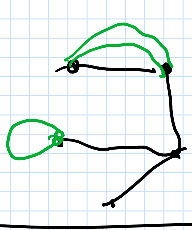
473 Lecture 4: Review of graph algorithms

9/2/21

$G=(V,E)$

$n=|V|$
$m=|E|$



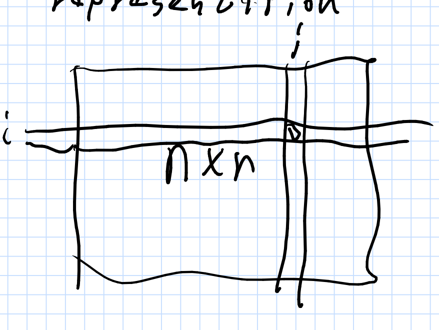
undirected graphs
simple graphs

directed graph

(u,v) directed edge
$u \to v$

uv $\{u,v\}$: undirected edge.

Matrix representation

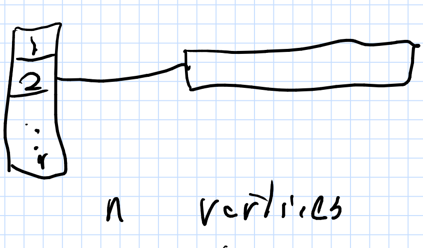


$V=[n]=\{1,2,..n\}$

$|E(G)|$ sparse graphs

$m \approx \tilde{O}(n)$

Adjacency list



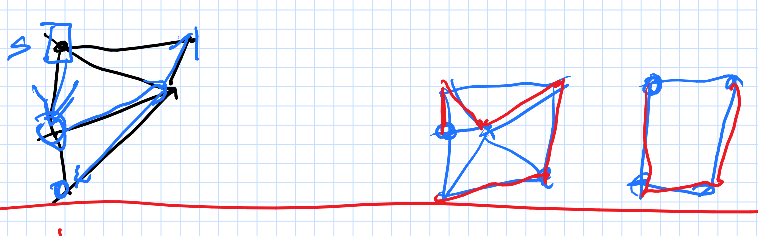
linked list of all its neighbors

n vertices m edges

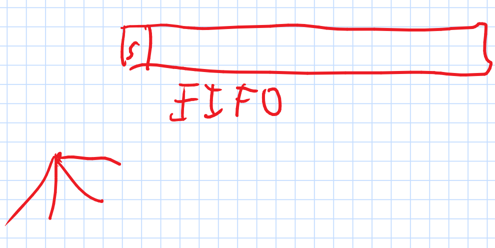
$O(n+m)$

Graph search/exploration

BFS or DFS   DFS depth first search

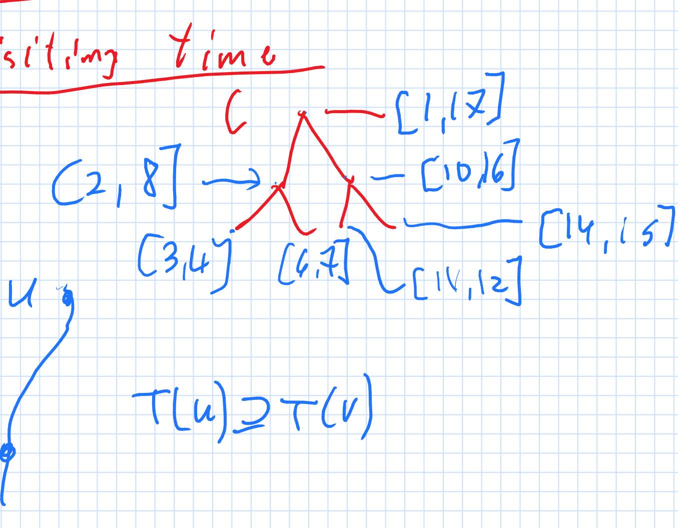


$O(n+m)$
BFS



FIFO   queue

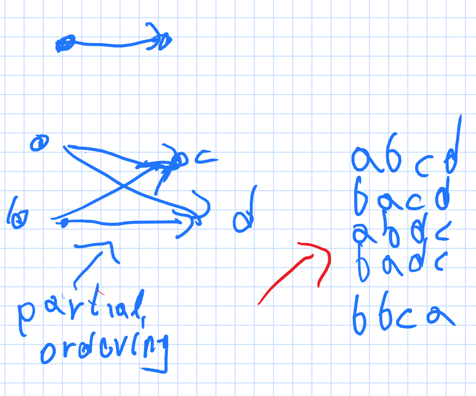
DFS visiting time

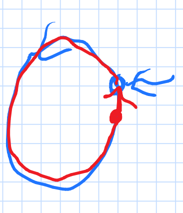


$T(u) \supseteq T(v)$

pre-order
post order: sort by time you leave a vertex.

DAG: Directed acyclic graph

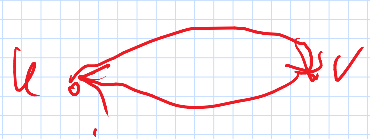




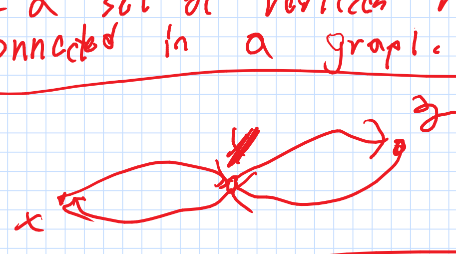
Topological ordering

Topological ordering ≡ decreasing post-ordering vertices

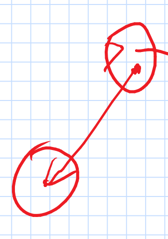
Strong connected component



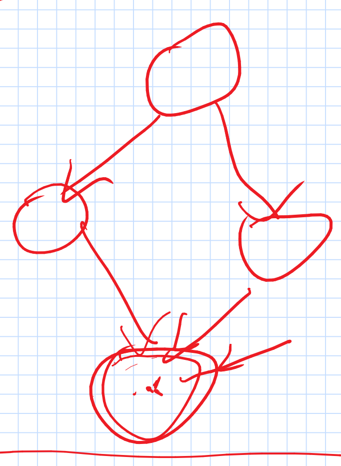
SCC = a maximum set of vertices that are strongly connected in a graph.



G^{SCC}

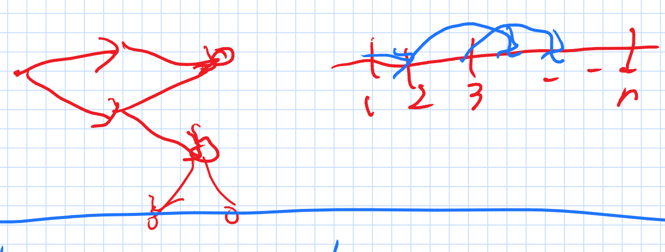


a simple SCC



$O(n+m)$

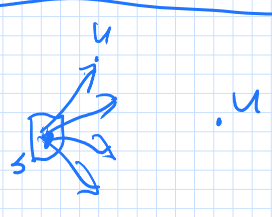
Topological ordering



Dijkstra ≡ Shortest path

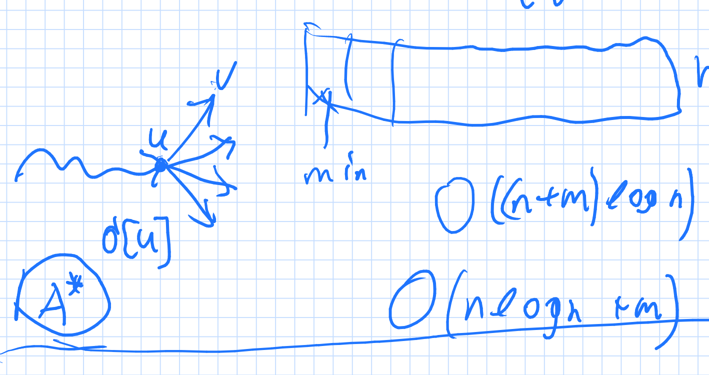
G directed
s: start vertex in G
weights on the edges

Compute the shortest path in G from s to all vertices.



$d[u]$ candidate shortest distance

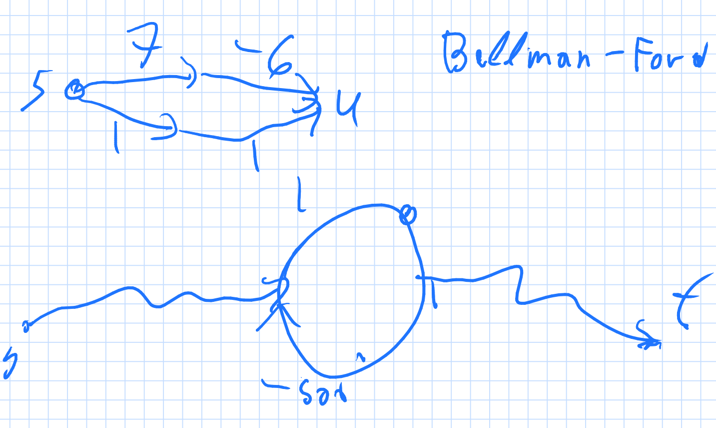
$d[s]=0$



heap
unvisited vertices

$O((n+m)\log n)$

A* $O(n\log n + m)$

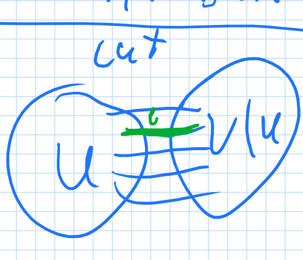


Bellman-Ford

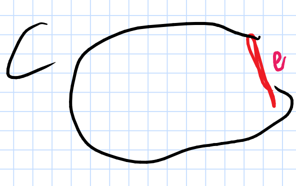
MST Minimum spanning trees

G: undirected, connected, positive weights on the edges

Compute minimum spanning tree of G   MST.



safe edge



unsafe edge

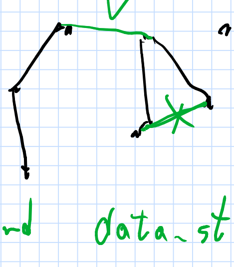
cost(e) is bigger than all other edges in the cycle.

prim algorithm

$O(n\log n + m)$ time

Kruskal algorithm

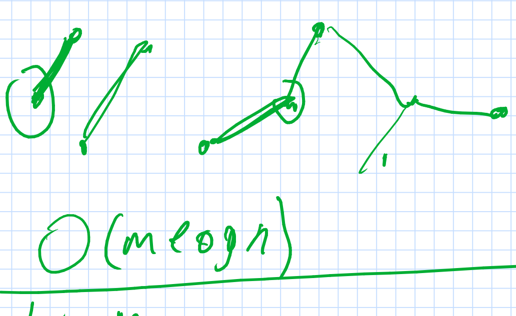
$e_1, e_2, \ldots, e_m$



union-find data-structure

$O(n+m\log m)$

Baruvka's algorithm 1926

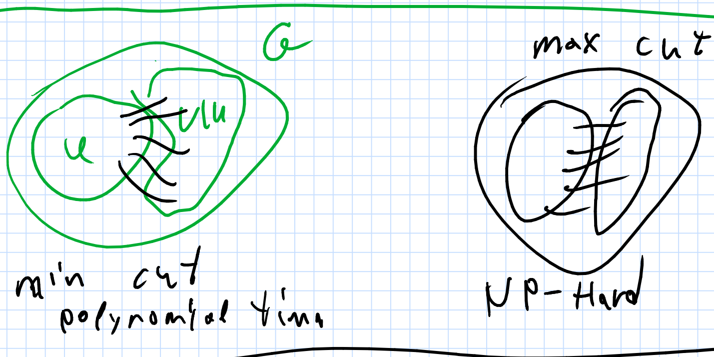


$O(m\log n)$

Directed MST

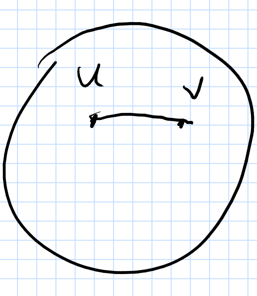
G directed graph
s: source vertex
Task: find the cheapest broadcast tree from s.

solved in polynomial time



Max 2SAT is NP-Hard

$(x_i \vee x_j) \wedge (\sim) \wedge (\sim) \ldots$



$x_u \in \{0,1\}$

$\bigwedge_{e\in E} (x_u \vee x_v) \wedge (\overline{x_u} \vee \overline{x_v})$