Lecture 5. Graphs Dec. 22. 2020

$G = (V, E)$

vertex set

$E \subseteq V \times V$

undirected	directed
$(u,v) \in E$	$(u,v) \in E$
$\Updownarrow$	$\not\Updownarrow$
$(v,u) \in E$	$(v,u) \in E$

adjacency matrix

$M_G = \begin{pmatrix} \\ \\ \end{pmatrix}$ ($V \times V$)

adjacency list

1: neighbors
2:
3:
⋮
n:

tradeoffs	$(u,v) \overset{?}{\in} E$	neighbor(v)	space
matrix	$O(1)$	$O(n)$	$O(n^2)$
list	$O(d)$	$O(d)$	$O(m)$

$n = |V|$

$m = |E|$

d: max neighbors.

Connectivity

depth-first search (DFS)

DFS(G):

$\forall v \in V$: visited[v] = false

$\forall v \in V$, if ~~visited~~ not visited[v]:

explore(G, v)

Explore(G, v)

goal: mark all vertices reachable from the start v.

visited[v] = true

for each $(v,u) \in E$:

if not visited[u]:

explore(G, u)

claim: Explore(G, v) visits every node reachable from v.

Proof: Suppose not. Let a be a vertex that is not visited and is reachable (By contradiction)

v(visited) — w(visited) — b(not) → a(not) b is the first unvisited.

Run time of DFS:

~~visit each~~ vertex once

consider every edge twice
(start and end)

$T(n) = O(n+m)$ global counting

Connectivity in undirected graphs:

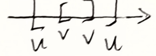
$G \to$ connected components

$u, v \in V$ are connected
if there is path from u to v.

Reading off CCs:

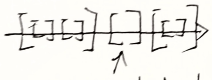
Types of intervals:

$(u,v) \in E$

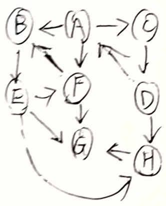


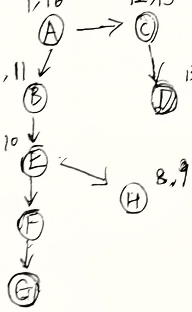
stack: u then v, v then u.

eg



3 connected components





CC(G):

$\forall v \in V$, visited[v] = false
ccnum[v] = null

cc := 0

$\forall v \in V$: if not visited[v]
cc++
explore(G,v)

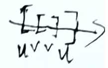
Explore(G,v):

visited[v] = true
ccnum[v] = cc
$\forall (v,u)$ in E:
if not visited[u]:
Explore(G,u)

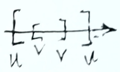
Types of edges

$(u,v) \in E$

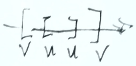
tree edge (solid)



forward edge



back edge



cross edge