Introduction to graph algorithms

Graph $G = (V, E)$

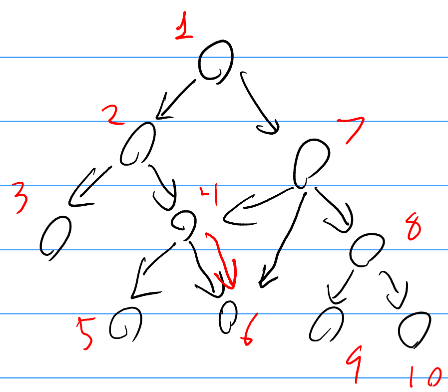
V = Set of vertices = # vertices
E = Set of edges = # edges

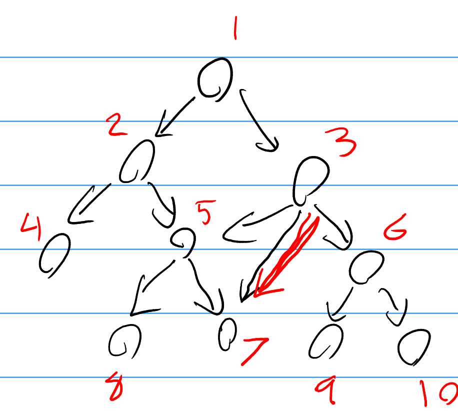
$E \leq \binom{V}{2}$ (undirected)
$\leq V(V-1)$ (directed)

Basic Question: Reachability

Given s, t find all reachable vertices
(maybe: & return path)

```
Visited = {}
def DFS(v):
    if v in visited: return
    visited.add(v)
    for w in v.adj:
        DFS(w)
```





DFS

by

```
def BFS(s, t):

    Q = queue([s])
    visited = {}
    while Q:
        v = Q.pop_front()
        if v in visited: continue
        visited.add(v)
        for w in v.adj:
            Q.push_back(w)
```

Pop_back

(and loop in reverse order to match DFS() but both are depth first)

BFS: queue

DFS: stack

min. spanning tree: heap / priority queue (Prim's Algorithm)

shortest paths: priority queue on different weights (Diskstra's Alg)

claim: whatever the pop() used, the whatever first search visits t $\Leftrightarrow \exists\ s \rightsquigarrow t$ path.

add Parent pointers to alg:

```
def BFS(s, t):

    Q = queue([(s, None)])
    Parent = {}
    while Q:
        v, P = Q.pop_front()
        if v in parent : continue
        Parent[v] = P
        for w in v.adj:
            Q.push_back((w, v))
```

Claim: Parent[v] exists at end $\iff$ v reachable from s, and if so, v → Parent[v] → ... ends at s.

a.k.a. v "visited"

Pf (v reachable ⟹ Parent[v] exists)

v reachable ⟹ ∃ path $s = u_1 \to u_2 \to u_3 \to \dots \to u_{k+1} = v$ from s to v.

We prove the claim ∀ vertices at distance k from s, by induction on k

Base case: $k = 0$.

Here, $v = s$. The algorithm sets Parent[s] = None, which exists.

Inductive step Suppose true for $k-1$.

Then ∀ v of distance k,

∃ path $s = u_1 \to u_2 \to \dots \to u_k \to u_{k+1} = v$

& u_k has distance $k-1$ ⟹ by inductive hypothesis it was visited.

When u_k was visited, since $(u_k, v) \in E$,

(v, u_k) was placed in Q.

$\Rightarrow$ (v, u_k) eventually removed from Q

$\Rightarrow$ v visited at some point

$\Rightarrow$ parent[v] set

$\Rightarrow$ inductive step true

$\Rightarrow$ claim holds $\forall k \geq 0$, as desired.

Pf ($v \to$ parent[v] $\to \ldots$ ends at s $\forall$ v "visited")

We prove this by induction on the # vertices visited.

Base case (1st vertex visited): $v = s$, parent[s] = None, ✓.

Inductive step: If v is the k^{th} vertex visited, for $k > 1$, u = parent(v) was visited earlier, so by the inductive hypothesis

$u \to$ parent[u] $\to$ parent(parent(u)) $\to \ldots$

ends at s.

$\Rightarrow$ $v \to$ parent(v) $\to \ldots$ ends at s.

$\Rightarrow$ inductive step $\Rightarrow$ induction holds $\forall k \Rightarrow$ claim.

Pf (v unreachable $\Rightarrow$ parent(v) does not exist at end)

The previous proves the contrapositive:

If parent(v) exists at end

$\Rightarrow$ $v \to$ parent(v) $\to \ldots$ ends at s

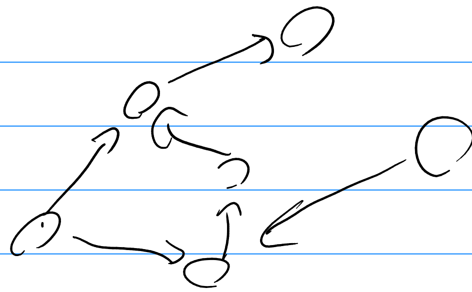
each of those is a reversed edge in E

$\Rightarrow$ $\exists$ $s \rightsquigarrow v$ path

$\Rightarrow$ v reachable.

Exercises

1) Road network:



weights = max height of trucks taking road (= min height of bridge)

Q: tallest truck that can go $s \to t$?

Q2: can go between any pair of locations?

2) Floodfill: MS Paint