Introduction to graph algorithms

Graph \( G = (V, E) \)

- \( V = \text{set of vertices} = \# \text{vertices} \)
- \( E = \text{set of edges} = \# \text{edges} \)

\[ E \leq \binom{V}{2} \quad \text{(undirected)} \]
\[ E \leq V(V-1) \quad \text{(directed)} \]

Basic Question: Reachability

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

\( \text{Visited} = \exists \delta \)

```python
def DFS(v):
    if v in visited: return
    visited.add(v)
    for w in v.adj:
        DFS(w)
```

![Graphs](image-url)
BFS: queue

DFS: stack

Claim: whatever the pop order visits $E \rightarrow F \rightarrow G \rightarrow H$. Paths shortest path to priority queue, then keep queue on different weights (Dijkstra's Alg).

while $Q$ is not empty:
  if $v$ in $V$ and $v$ not visited:
    $v$ push to back of $Q$ (for recall)
    for $w$ in $v$ adjacency:
      if $w$ not visited:
        $w$ push to front of $Q$
        $v$ visited

$Q = \text{queue}(E,F,G,J,H)$

visited = $E,F,G,J,H$
add parent pointers to algo:

```python
def BFS(S, E):
    Q = queue([E(S, None)])
    Parent = {}
    while Q:
        P = Q.pop_front()
        if P in parent: continue
        Parent[P] = P
        for v in P.adj:
            Q.push_back((v, P))
```

Claim: Parent[P] exists at end if a.u.a. V visited
V reachable from S,
and if so, \( v \to P \text{ in } V \to P \) ends at S.

\[ \text{if } (V \text{ reachable } \supset \text{ Parent}[V] \text{ exists }) \]

V reachable \( \supset \) \( \exists \) path \( S = u_1 \to u_2 \to u_3 \to \ldots \to u_k = v \)
from S to V.

We prove the claim if vertices at distance \( k \) from S, by induction on k.

Base case: \( k = 0 \).
Here, \( V = S \). The algorithm sets
\[ \text{Parent}[S] = \text{None}, \text{ which exists.} \]

Inductive step: Suppose true for \( k - 1 \).
Then \( \forall \; v \) of distance \( k \),
\( \exists \) path \( S = u_1 \to u_2 \to \ldots \to u_k \to u_{k+1} = v \)
& \( u_k \) has distance \( k - 1 \) \( \supset \) by inductive
hypothesis it was visited.
When \( u_k \) was visited, since \((u_k, v) \in E\),
\((v, u_n)\) was placed in \(Q\).
\[ \Rightarrow (v, u_n) \text{ eventually removed from } Q \]
\[ \Rightarrow v \text{ visited at some point} \]
\[ \Rightarrow \text{parent}(v) \text{ set} \]
\[ \Rightarrow \text{inductive step true} \]
\[ \Rightarrow \text{claim holds } \forall k \geq 0, \text{ as desired.} \]

\[ \text{Pr ( } v \rightarrow \text{parent}(v) \rightarrow \ldots \text{ ends at } s \]
\[ \forall v \text{ "visited"}) \]

We prove this by induction on the \# vertices visited.

Base case (1st vertex visited): \(v = s\), parent\((s)\) = None, \(\checkmark\).

Inductive step: If \(v\) is the \(k\)th vertex visited,

for \(k > 1\), \(u = \text{parent}(v)\) was visited earlier,

so by the inductive hypothesis

\[ u \rightarrow \text{parent}(u) \rightarrow \text{parent}(\text{parent}(u)) \rightarrow \ldots \text{ends at } s \]

\[ \Rightarrow v \rightarrow \text{parent}(v) \rightarrow \ldots \text{ ends at } s \]

\[ \Rightarrow \text{inductive step} \Rightarrow \text{induction holds } \forall k \geq 0 \text{ claim.} \]

\[ \text{Pr ( } v \text{ unreachable } \Rightarrow \text{parent}(v) \text{ does not exist at end) } \]

The previous proves the contrapositive:

\[ \text{if } \text{parent}(v) \text{ exists at end} \]
\[ \Rightarrow v \rightarrow \text{parent}(v) \rightarrow \ldots \text{ ends at } s \]

each of those is a reverse edge in \(E\)

\[ \Rightarrow \exists s \rightarrow v \text{ path} \]
\[ \equiv v \text{ reachable.} \]
Exercises

1) Road network: weights = max height of trucks taking road/1 = min height of bridge

Q1: tallest truck that can go S→T?
Q2: can go between any pair of locations?

2) Floodfill: MS Paint