Lecture 14-A

Graph Searching: Breadth-First Search
Graph Searching: Breadth-First Search

Graph $G = (V, E)$, directed or undirected with adjacency list representation.

**GOAL:** Systematically explores edges of $G$ to

- discover every vertex reachable from the source vertex $s$
- compute the shortest path distance of every vertex from the source vertex $s$
- produce a breadth-first tree (BFT) $G_\Pi$ with root $s$
  - BFT contains all vertices reachable from $s$
  - the unique path from any vertex $v$ to $s$ in $G_\Pi$ constitutes a shortest path from $s$ to $v$ in $G$

**IDEA:** Expanding frontier across the breadth-greedy-

- propagate a wave 1 edge-distance at a time
- using a FIFO queue: $O(1)$ time to update pointers to both ends
Breadth-First Search Algorithm

Maintains the following fields for each \( u \in V \)
- \( \text{color}[u] \): color of \( u \)
  - WHITE : not discovered yet
  - GRAY : discovered and to be or being processed
  - BLACK: discovered and processed
- \( \Pi[u] \): parent of \( u \) (NIL of \( u = s \) or \( u \) is not discovered yet)
- \( d[u] \): distance of \( u \) from \( s \)

Processing a vertex = scanning its adjacency list
Breadth-First Search Algorithm

BFS\((G, s)\)

```plaintext
for each \(u \in V - \{s\}\) do
    color\([u]\) \leftarrow \text{WHITE}
    \Pi[u] \leftarrow \text{NIL}; d [u] \leftarrow \infty

color\([s]\) \leftarrow \text{GRAY}
\Pi[s] \leftarrow \text{NIL}; d [s] \leftarrow 0

Q \leftarrow \{s\}
while \(Q \neq \emptyset\) do
    u \leftarrow \text{head}[Q]
    for each \(v\) in Adj\([u]\) do
        if color\([v]\) = \text{WHITE} then
            color\([v]\) \leftarrow \text{GRAY}
            \Pi[v] \leftarrow u
            d [v] \leftarrow d [u] + 1
            ENQUEUE\(Q, v\)

DEQUEUE\(Q\)
    color\([u]\) \leftarrow \text{BLACK}
```
Breadth-First Search

Sample Graph:

FIFO queue $Q$ just after processing vertex $\langle a \rangle$
Breadth-First Search

```plaintext
FIFO queue Q
just after processing vertex

\{a\}
\{a,b,c\}

s
0
a
1
b
c
d
e
f
h
i
g

1

- a
```
Breadth-First Search

- Breadth-first search is a graph traversal algorithm that explores the nodes of a graph in a breadthward motion.

- The algorithm starts at the root node (in this case, node s) and explores all the neighboring nodes at the current depth before moving on to the nodes at the next depth level.

- Example:
  - Initial queue: \(\langle a \rangle\)  
  - Just after processing vertex \(a\): queue \(Q\) changes to \(\langle a, b, c \rangle\)  
  - Just after processing vertex \(b\): queue \(Q\) changes to \(\langle a, b, c, f \rangle\)
Breadth-First Search

FIFO queue $Q$ just after processing vertex

- $\langle a \rangle$
- $\langle a,b,c \rangle$
- $\langle a,b,c,f \rangle$
- $\langle a,b,c,f,e \rangle$

- a
- b
- c
Breadth-First Search

FIFO queue just after processing vertex

\[
\begin{align*}
\langle a \rangle & \quad \text{-} \\
\langle a, b, c \rangle & \quad a \\
\langle a, b, c, f \rangle & \quad b \\
\langle a, b, c, f, e \rangle & \quad c \\
\langle a, b, c, f, e, g, h \rangle & \quad f
\end{align*}
\]
Breadth-First Search

FIFO just after queue $Q$ processing vertex

\[
\begin{align*}
\langle a \rangle &\quad - \\
\langle a,b,c \rangle &\quad a \\
\langle a,b,c,f \rangle &\quad b \\
\langle a,b,c,f,e \rangle &\quad c \\
\langle a,b,c,f,e,g,h \rangle &\quad f \\
\langle a,b,c,f,e,g,h,d,i \rangle &\quad e \\
\end{align*}
\]

all distances are filled in after processing $e$
Breadth-First Search

FIFO queue $Q$ just after processing vertex:

- $\langle a \rangle$
- $\langle a,b,c \rangle$  
- $\langle a,b,c,f \rangle$  
- $\langle a,b,c,f,e \rangle$  
- $\langle a,b,c,f,e,g,h \rangle$  
- $\langle a,b,c,f,e,g,h,d,i \rangle$
Breadth-First Search

FIFO just after queue $Q$ processing vertex

$$
\langle a \rangle \\
\langle a,b,c \rangle \quad a \\
\langle a,b,c,f \rangle \quad b \\
\langle a,b,c,f,e \rangle \quad c \\
\langle a,b,c,f,e,g,h \rangle \quad f \\
\langle a,b,c,f,e,g,h,d,i \rangle \quad h
$$
Breadth-First Search

FIFO just after processing vertex

\begin{itemize}
  \item \langle a \rangle
  \item \langle a,b,c \rangle \quad a
  \item \langle a,b,c,f \rangle \quad b
  \item \langle a,b,c,f,e \rangle \quad c
  \item \langle a,b,c,f,e,g,h \rangle \quad f
  \item \langle a,b,c,f,e,g,h,d,i \rangle \quad d
\end{itemize}
Breadth-First Search

FIFO queue just after processing vertex

\[
\begin{align*}
\langle a \rangle & \quad - \\
\langle a, b, c \rangle & \quad a \\
\langle a, b, c, f \rangle & \quad b \\
\langle a, b, c, f, e \rangle & \quad c \\
\langle a, b, c, f, e, g, h \rangle & \quad f \\
\langle a, b, c, f, e, g, h, d, i \rangle & \quad i
\end{align*}
\]

algorithm terminates: all vertices are processed
Breadth-First Search Algorithm

Running time: $O(V+E) = \text{considered linear time in graphs}$

- initialization: $\Theta(V)$
- queue operations: $O(V)$
  - each vertex enqueued and dequeued at most once
  - both enqueue and dequeue operations take $O(1)$ time
- processing gray vertices: $O(E)$
  - each vertex is processed at most once and
  $$\sum_{u \in V} |\text{Adj}[u]| = \Theta(E)$$
Theorems Related to BFS

DEF: $\delta(s, v) =$ shortest path distance from $s$ to $v$

LEMMA 1: for any $s \in V$ & $(u, v) \in E$; $\delta(s, v) \leq \delta(s, u) + 1$

For any BFS($G, s$) run on $G=(V,E)$

LEMMA 2: $d[v] \geq \delta(s, v)$ $\forall v \in V$

LEMMA 3: at any time of BFS, the queue $Q=\langle v_1, v_2, \ldots, v_r \rangle$ satisfies

- $d[v_r] \leq d[v_1] + 1$
- $d[v_i] \leq d[v_{i+1}]$, for $i = 1, 2, \ldots, r - 1$

THM1: BFS($G, s$) achieves the following

- discovers every $v \in V$ where $s \rightarrow v$ (i.e., $v$ is reachable from $s$)
- upon termination, $d[v] = \delta(s, v)$ $\forall v \in V$
- for any $v \neq s$ & $s \rightarrow v$; $sp(s, \Pi[v]) \sim (\Pi[v], v)$ is a sp($s, v$)
Proofs of BFS Theorems

**DEF:** shortest path distance $\delta(s, v)$ from $s$ to $v$

$\delta(s, v) = \text{minimum number of edges in any path from } s \text{ to } v$

$= \infty$ if no such path exists (i.e., $v$ is not reachable from $s$)

**L1:** for any $s \in V \& (u, v) \in E; \delta(s, v) \leq \delta(s, u) + 1$

**PROOF:** $s \rightarrow u \Rightarrow s \rightarrow v$. Then,

consider the path $p(s, v) = sp(s, u) \sim (u, v)$

- $|p(s, v)| = |sp(s, u)| + 1 = \delta(s, u) + 1$
- therefore, $\delta(s, v) \leq |p(s, v)| = \delta(s, u) + 1$
Proofs of BFS Theorems

**DEF:** shortest path distance $\delta(s, v)$ from $s$ to $v$

\[ \delta(s, v) = \text{minimum number of edges in any path from } s \text{ to } v \]

**L1:** for any $s \in V$ & $(u, v) \in E$; $\delta(s, v) \leq \delta(s, u) + 1$

**C1 of L1:** if $G=(V,E)$ is undirected then $(u, v) \in E \Rightarrow (v, u) \in E$

- $\delta(s, v) \leq \delta(s, u) + 1$ and $\delta(s, u) \leq \delta(s, v) + 1$
- $\Rightarrow \delta(s, u) - 1 \leq \delta(s, v) \leq \delta(s, u) + 1$ and $\delta(s, v) - 1 \leq \delta(s, u) \leq \delta(s, v) + 1$
- $\Rightarrow \delta(s, u) \& \delta(s, v)$ differ by at most 1
Proofs of BFS Theorems

L2: upon termination of BFS\((G, s)\) on \(G=(V,E)\);
\[d[v] \geq \delta(s, v) \quad \forall v \in V\]

PROOF: by induction on the number of ENQUEUE operations
- **basis:** immediately after 1st enqueue operation
  \(\text{ENQ}(Q, s): d[s] = \delta(s, s)\)
- **hypothesis:** \(d[v] \geq \delta(s, v)\) for all \(v\) inserted into \(Q\)
- **induction:** consider a white vertex \(v\) discovered during scanning \(\text{Adj}[u]\)
  \[d[v] = d[u] + 1\] due to the assignment statement
  \[\geq \delta(s, u) + 1\] due to the inductive hypothesis since \(u \in Q\)
  \[\geq \delta(s, v)\] due to L1
- **vertex** \(v\) is then enqueued and it is never enqueued again
  \(d[v]\) never changes again, maintaining inductive hypothesis
Proofs of BFS Theorems

L3: Let $Q = \langle v_1, v_2, \ldots, v_r \rangle$ during the execution of BFS($G, s$), then, $d[v_r] \leq d[v_1] + 1$ and $d[v_i] \leq d[v_{i+1}]$ for $i = 1, 2, \ldots, r-1$

PROOF: by induction on the number of QUEUE operations

- **basis:** lemma holds when $Q \leftarrow \{s\}$
- **hypothesis:** lemma holds for a particular $Q$ (i.e., after a certain # of QUEUE operations)
- **induction:** must prove lemma holds after both DEQUEUE & ENQUEUE operations

- **DEQUEUE($Q$):** $Q = \langle v_1, v_2, \ldots, v_r \rangle \Rightarrow Q' = \langle v_2, v_3, \ldots, v_r \rangle$
  
  \begin{align*}
  - d[v_r] & \leq d[v_1] + 1 \quad \& d[v_1] \leq d[v_2] \quad \text{in } Q \Rightarrow \\
  - d[v_i] & \leq d[v_{i+1}] \quad \text{for } i = 1, 2, \ldots, r-1 \quad \text{in } Q \Rightarrow \\
  d[v_i] & \leq d[v_{i+1}] \quad \text{for } i = 2, \ldots, r-1 \quad \text{in } Q'
  \end{align*}
Proofs of BFS Theorems

- **ENQUEUE** \((Q, v)\): 
  \[ Q = \langle v_1, v_2, \ldots, v_r \rangle \Rightarrow Q' = \langle v_1, v_2, \ldots, v_r, v_{r+1} = v \rangle \]
  
  - \(v\) was encountered during scanning \(\text{Adj}[u]\) where \(u = v_1\)
  
  - thus, \(d[v_{r+1}] = d[v] = d[u] + 1 = d[v_1] + 1 \Rightarrow d[v_{r+1}] = d[v_1] + 1\) in \(Q'\)
  
  - but \(d[v_r] \leq d[v_1] + 1 = d[v_{r+1}]\)
  
  - \(\Rightarrow d[v_{r+1}] = d[v_1] + 1\) and \(d[v_r] \leq d[v_{r+1}]\) in \(Q'\)

C3 of L3 (monotonicity property):

- **if** the vertices are enqueued in the order \(v_1, v_2, \ldots, v_n\)
  
  - **then** the sequence of distances is monotonically increasing, i.e., \(d[v_1] \leq d[v_2] \leq \ldots \ldots \leq d[v_n]\)
Proofs of BFS Theorems

THM (correctness of BFS): BFS($G, s$) achieves the following on $G=(V,E)$

- discovers every $v \in V$ where $s \rightarrow v$
- upon termination: $d[v] = \delta(s, v) \quad \forall v \in V$
- for any $v \neq s$ & $s \rightarrow v$; $sp(s, \Pi[v]) \sim (\Pi[v], v) = sp(s, v)$

PROOF: by induction on $k$, where $V_k = \{v \in V: \delta(s, v) = k\}$

- hypothesis: for each $v \in V_k$, $\exists$ exactly one point during execution of BFS at which $\text{color}[v] \leftarrow \text{GRAY}$, $d[v] \leftarrow k$, $\Pi[v] \leftarrow u \in V_{k-1}$, and then $\text{ENQUEUE}(Q, v)$
- basis: for $k = 0$ since $V_0 = \{s\}$; $\text{color}[s] \leftarrow \text{GRAY}$, $d[s] \leftarrow 0$ and $\text{ENQUEUE}(Q, s)$
- induction: must prove hypothesis holds for each $v \in V_{k+1}$
Proofs of BFS Theorems

Consider an arbitrary vertex \( v \in V_{k+1} \), where \( k \geq 0 \)

- monotonicity (L3) + \( d[v] \geq k + 1 \) (L2) + inductive hypothesis
  \( \Rightarrow v \) must be discovered after all vertices in \( V_k \) were enqueued
- since \( \delta(s, v) = k + 1 \), \( \exists u \in V_k \) such that \( (u, v) \in E \)
- let \( u \in V_k \) be the first such vertex grayed (must happen due to hyp.)
- \( u \leftarrow \text{head}(Q) \) will be ultimately executed since BFS enqueues every grayed vertex
  - \( v \) will be discovered during scanning \( \text{Adj}[u] \)
  - color\[v]\=WHITE since \( v \) isn’t adjacent to any vertex in \( V_j \) for \( j<k \)
  - color\[v]\ \leftarrow \text{GRAY}, d[v] \leftarrow d[u] + 1, \Pi[v] \leftarrow u
  - then, \( \text{ENQUEUE}(Q, v) \) thus proving the inductive hypothesis

To conclude the proof

- if \( v \in V_{k+1} \) then due to above inductive proof \( \Pi[v] \in V_k \)
  - thus \( \text{sp}(s, \Pi[v]) \sim (\Pi[v], v) \) is a shortest path from \( s \) to \( v \)
Theorems Related to BFS

DEF: \( \delta(s, v) = \) shortest path distance from \( s \) to \( v \)

LEMMA 1: for any \( s \in V \) & \( (u, v) \in E; \) \( \delta(s, v) \leq \delta(s, u) + 1 \)

For any BFS\((G, s)\) run on \( G=(V,E) \)

LEMMA 2: \( d[v] \geq \delta(s, v) \) \( \forall v \in V \)

LEMMA 3: at any time of BFS, the queue \( Q=\langle v_1, v_2, \ldots, v_r \rangle \) satisfies

- \( d[v_r] \leq d[v_1] + 1 \)
- \( d[v_i] \leq d[v_{i+1}] \), for \( i = 1, 2, \ldots, r - 1 \)

THM1: BFS\((G, s)\) achieves the following

- discovers every \( v \in V \) where \( s \rightarrow v \) (i.e., \( v \) is reachable from \( s \))
- upon termination, \( d[v] = \delta(s, v) \) \( \forall v \in V \)
- for any \( v \neq s \) & \( s \rightarrow v; \) sp\((s, \Pi[v]) \sim (\Pi[v], v)\) is a sp\((s, v)\)
LEMMA 4: predecessor subgraph $G_{\Pi}=(V_{\Pi}, E_{\Pi})$ generated by BFS($G, s$), where $V_{\Pi} = \{v \in V: \Pi[v] \neq \text{NIL}\} \cup \{s\}$ and $E_{\Pi} = \{(\Pi[v], v) \in E: v \in V_{\Pi} - \{s\}\}$ is a breadth-first tree such that

- $V_{\Pi}$ consists of all vertices in $V$ that are reachable from $s$
- $\forall v \in V_{\Pi}$, unique path $p(v, s)$ in $G_{\Pi}$ constitutes a $sp(s, v)$ in $G$

**PRINT-PATH**($G, s, v$)

```plaintext
if $v = s$ then print $s$
else if $\Pi[v] = \text{NIL}$ then
    print no “$s \rightarrow v$ path”
else
    **PRINT-PATH**($G, s, \Pi[v]$)
    print $v$
```

Prints out vertices on a $s \rightarrow v$ shortest path
Breadth-First Tree Generated by BFS

BFS(G,a) terminated

BFT generated by BFS(G,a)