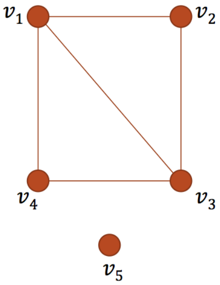
CSE101: Design and Analysis of Algorithms

Ragesh Jaiswal, CSE, UCSD

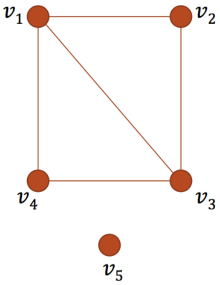
Graphs

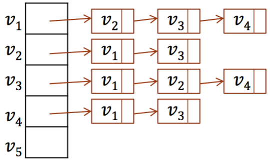
- Adjacency matrix: Store connectivity in a matrix.
- Space: $O(n^2)$



	v_1	v_2	v_3	v_4	v_5
v_1	0	1	1	1	0
v_2	1	0	1	0	0
v_3	1	1	0	1	0
v_4	1	0	1	0	0
v_5	0	0	0	0	0

- Adjacency list: For each vertex, store its neighbors.
- Space: $O(n + m)$





Graph Algorithms

Problem

Given an (undirected) graph $G = (V, E)$ and two vertices s, t, check if there is a path between s and t.

Problem

Given an (undirected) graph $G = (V, E)$ and two vertices s, t, check if there is a path between s and t.

- Alternate problem: What are the vertices that are reachable from s. Is t among these reachable vertices?
- This is also known as *graph exploration*. That is, explore all vertices reachable from a starting vertex s.
  - Breadth First Search (BFS)
  - Depth First Search (DFS)

Breadth First Search (BFS)

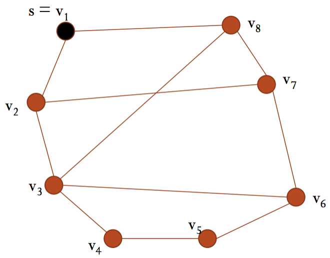
BFS(G, s)

- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
  - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
  - Put these nodes in the set $Layer(i)$
  - If $Layer(i)$ is empty, then end
  - $i \leftarrow i + 1$

Breadth First Search (BFS)

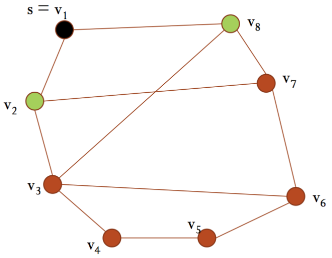
BFS(G, s)

- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
    - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
    - Put these nodes in the set $Layer(i)$
    - If $Layer(i)$ is empty, then end
    - $i \leftarrow i + 1$



Breadth First Search (BFS)

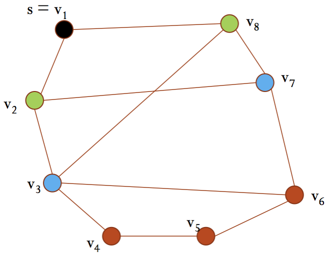
BFS(G, s)
- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
  - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
  - Put these nodes in the set $Layer(i)$
  - If $Layer(i)$ is empty, then end
  - $i \leftarrow i+1$



Breadth First Search (BFS)

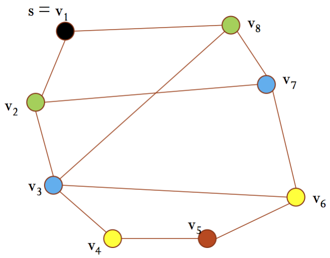
BFS(G, s)

- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
  - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
  - Put these nodes in the set $Layer(i)$
  - If $Layer(i)$ is empty, then end
  - $i \leftarrow i+1$



Breadth First Search (BFS)

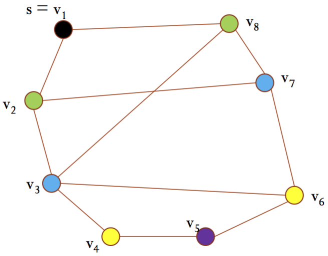
BFS(G, s)
- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
  - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
  - Put these nodes in the set $Layer(i)$
  - If $Layer(i)$ is empty, then end
  - $i \leftarrow i+1$



Breadth First Search (BFS)

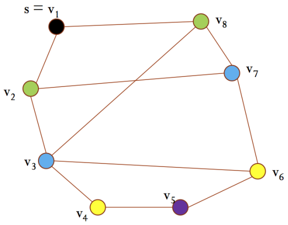
BFS(G, s)

- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
  - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
  - Put these nodes in the set $Layer(i)$
  - If $Layer(i)$ is empty, then end
  - $i \leftarrow i + 1$



Breadth First Search (BFS)

BFS(G, s)
- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
  - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
  - Put these nodes in the set $Layer(i)$
  - If $Layer(i)$ is empty, then end
  - $i \leftarrow i+1$



- Theorem 1: The shortest path from s to any vertex in $Layer(i)$ is equal to i.

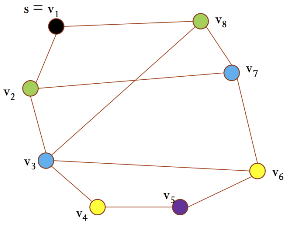
- Theorem 1: The shortest path from s to any vertex in $Layer(i)$ is equal to i.

Proof sketch

- We will prove by induction. Let $P(i)$ denote the statement: *The shortest path from s to any vertex in $Layer(i)$ is equal to i.*
- We will prove that $P(i)$ is true for all i using induction.
- Base case: $P(0)$ is true since $Layer(0)$ contains s.
- Inductive step: Assume $P(0), ..., P(k)$ are true. We will show that $P(k+1)$ is true.
  - Assume for the sake of contradiction that $P(k+1)$ is not true.
  - This implies that there is a vertex v in $Layer(k+1)$ such that the shortest path length from s to v is $< k+1$ (the case $> k+1$ is skipped for class discussion)
  - Consider such a path from s to v. Let u be the vertex in this path just before v.
  - Claim 1: u is contained in $Layer(k)$.
  - This gives us a contradiction since by induction hypothesis, the shortest path length from s to u is k.

Breadth First Search (BFS)

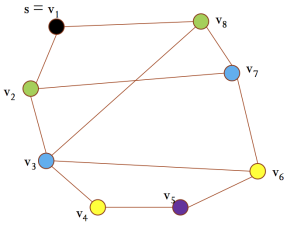
BFS(G, s)
- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
  - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
  - Put these nodes in the set $Layer(i)$
  - If $Layer(i)$ is empty, then end
  - $i \leftarrow i + 1$



- What is the running time of BFS given that the graph is given in adjacency list representation?

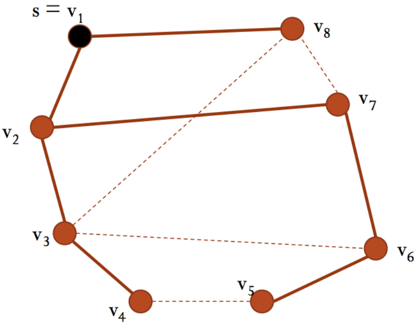
Breadth First Search (BFS)

BFS(G, s)
- $Layer(0) = \{s\}$
- $i \leftarrow 1$
- While(true)
    - Visit all new nodes that have an edge to a vertex in $Layer(i-1)$
    - Put these nodes in the set $Layer(i)$
    - If $Layer(i)$ is empty, then end
    - $i \leftarrow i + 1$



- What is the running time of BFS given that the graph is given in adjacency list representation? $O(n+m)$

- The BFS algorithm defines the following BFS tree rooted at s
  - Vertex u is the parent of vertex v if u caused the immediate discovery of v.



End