Connected Components and Correctness of BFS in SSSP

CSCI 2100 Teaching Team

Outline

Today's tutorial covers:

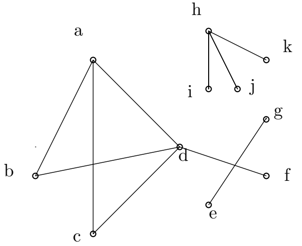
- finding connected components;
- proving that BFS correctly solves the unit-weight SSSP problem.

Connected Components

Problem: Let $G = (V, E)$ be an undirected graph. Our goal is to compute all the **connected components** (CC) of G.

A CC of G includes a set $S \subseteq V$ of vertices such that:

- (Connectivity) any two vertices in S are reachable from each other;
- (Maximality) it is not possible to add another vertex to S while still satisfying the above requirement.



G

Output:
$\{a, b, c, d, f\}, \{g, e\}, \{h, i, j, k\}$

A Lemma on CCs

Lemma 1: Take an arbitrary vertex s. The CC of s is the set S of vertices in G reachable from s.

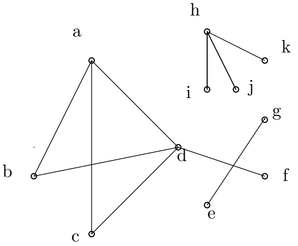
Proof:

- Connectivity: any two vertices in S can reach each other via s.
- Maximality: any vertex outside S is unreachable from s.

□

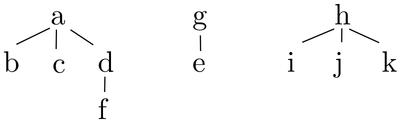
A BFS Solution

1. Run BFS on G starting from a white source vertex
2. Output the vertex set of the BFS-tree
3. If there is still a white vertex in G, repeat from 1



G

BFS-forest



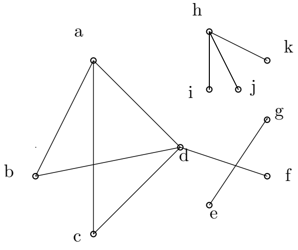
Proof of Correctness

Claim: The vertex set S of every BFS-tree is a CC of G.

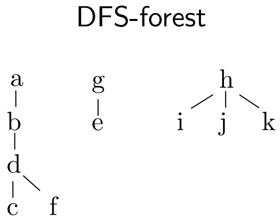
Proof: Follows immediately because BFS finds all the vertices reachable from s. $\square$

A DFS Solution

1. Run DFS on G starting from a white source vertex
2. Output the vertex set of the DFS-tree
3. If there is still a white vertex in G, repeat from 1



G



Proof of correctness

Claim: The vertex set S of each DFS-tree is a CC of G.

Proof: Let s be the source vertex of DFS. We will show that the DFS-tree contains all and only the vertices reachable from s.

Let v be a vertex reachable from s. At the beginning of DFS, there is a white path from s to v. By the white path theorem, v must be in the subtree of s, namely, in the DFS-tree.

It is obvious that every vertex in the DFS-tree is reachable from s.

□

Single Source Shortest Path (SSSP) with Unit Weights

Let $G = (V, E)$ be a directed graph, and s be a vertex in V. The goal of the **SSSP problem** is to find, for every other vertex $t \in V \setminus \{s\}$, a shortest path from s to t, unless t is unreachable from s.

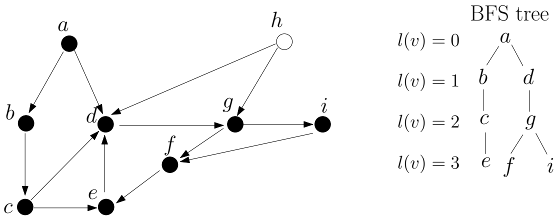
Using BFS to Solve SSSP Problem

Run BFS algorithm starting from s on G, which returns a **BFS-tree** T.

For any $v \in V \setminus \{s\}$, report the path from s to v in T as the shortest path from s to v in G.

Proof of Correctness

We first prove a useful lemma.



Lemma 2: For any two vertices $u, v \in V$ such that $u \neq v$, if $l(u) < l(v)$, it must hold that u is enqueued before v during the BFS.

Proof: We will prove this by induction.

Base Case. $l(u) < l(v) \leq 1$.

We must have $l(u) = 0$ and $l(v) = 1$, which means u is the source s. As s is enqueued at the very beginning of BFS, s is enqueued before v. The base case holds.

Inductive Case.

Inductive assumption: For any two vertices u, v satisfying $l(u) < l(v) \leq L - 1$ where $L \geq 2$, it always holds that u is enqueued before v.

Consider any vertices u and v satisfying $l(u) < l(v) = L$. Let p_u and p_v be their parents in the BFS-tree T, respectively. We have $l(p_u) = l(u) - 1$ and $l(p_v) = l(v) - 1$.

It follows that $l(p_u) < l(p_v) \leq L - 1$. By the inductive assumption, p_u is enqueued before p_v. From the FIFO property of queue, p_u is dequeued before p_v. As u (resp., v) is enqueued right after p_u (resp., p_v) is dequeued, u is enqueued before v.

□

We now prove the correctness of BBS.

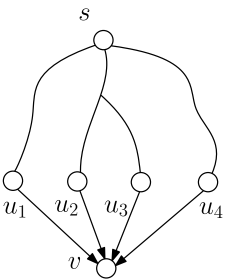
Theorem: For any vertex $v \in V$, the path from s to v in T is a shortest path from s to v in G.

We will prove a stronger claim by induction:

Claim: If a vertex $v \in V$ has shortest path distance L from s, then $l(v) = L$.

Base Case. $L = 0$ or 1.

- s is the only vertex with shortest path distance 0 from s. It is obvious that $l(s) = 0$.
- Every vertex v with shortest path distance 1 from s will be enqueued when s is dequeued and thus has $l(v) = 1$.

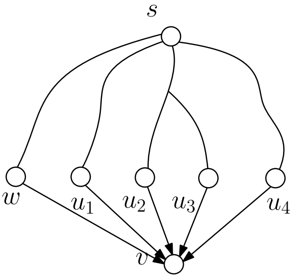


Inductive Case.

Inductive assumption: If a vertex v has shortest path distance $L \leq k - 1$ from s where $k \geq 2$, then $l(v) = L$.

Let v be a vertex with shortest path distance k from s. Consider all the shortest paths from s to v and let U denote the set of predecessors of v in those paths. Furthermore, let u_1 denote the vertex in U that was enqueued the earliest. The shortest path distance from s to u_1 is $k - 1$.

By the induction assumption, $l(u_1) = k - 1$. To prove $l(v) = k$, it suffices to prove that v is enqueued at the moment u_1 is dequeued, namely, v is still white when u_1 is dequeued. We will prove this by contradiction.



Suppose that when u_1 is dequeued, v is not white.

Define w as the parent of v in T (i.e., v is enqueued after w is dequeued). By Lemma 2, We have $l(w) \leq l(u_1)$ as w is dequeued before u_1. We further have $l(w) \neq l(u_1)$; otherwise, w must belong to U, which contradicts the definition of u_1.

It follows that $l(w) < l(u_1)$. However, this means that the shortest path distance from s to w is less than $k - 1$. Thus, the shortest path distance from s to v is less than k, giving a contradiction.

□