Lecture 9

Instructor: Karteek Sreenivasaiah

14th September 2018

Recap

Algorithm 1 Breadth-first Search from vertex s

Color all vertices WHITE.
For all $u \in V$, $d[u] \leftarrow \infty$, $\pi[u] \leftarrow$ NIL.
$d[s] \leftarrow 0$.
Initialize queue $Q \leftarrow \emptyset$.
ENQUEUE(Q, s)
while $Q \neq \emptyset$ **do**
    $u \leftarrow$ DEQUEUE(Q)
    for each $v \in \mathcal{N}(u)$ **do**
        if color(v) =WHITE **then**
            color$[v] \leftarrow$ GRAY
            $d[v] \leftarrow d[u] + 1$
            $\pi[v] \leftarrow u$
            ENQUEUE(Q, v)
        end if
    end for
    color$[u] \leftarrow$ BLACK.
end while

Correctness of BFS

Notation: Let $\delta(s, v)$ denote the minimum number of edges on a path from s to v.

Theorem

Let $G = (V, E)$ be a graph. When BFS is run on G from vertex $s \in V$:

1. Every vertex that is reachable from s gets discovered.
2. On termination, $d[v] = \delta(s, v)$.

Show (1) is an exercise.

Proof of correctness

Proof

Suppose, for the sake of contradiction, (2) does not hold.
Let v be the vertex with smallest $\delta(s, v)$ such that $d[v] \neq \delta(s, v)$.
Claim 1: $d[v] \geq \delta(s, v)$
Let u be the vertex just before v on any path from s to v.
Claim 2: $\delta(s, v) \leq \delta(s, u) + 1$.
Choose a *shortest* path from s to v.
Let u be the vertex immediately preceding v
Then $\delta(s, v) = \delta(s, u) + 1 = d[u] + 1$.
So we have:

$$d[v] > \delta(s, v) = \delta(s, u) + 1 = d[u] + 1$$

Proof of correctness

Proof cont...

We have:

$$d[v] > \delta(s, v) = \delta(s, u) + 1 = d[u] + 1$$

Consider the time step when u is dequeued.

- Case 1: v was white.
  The algo sets $d[v] = d[u] + 1$.
  This contradicts the eq above.
- Case 2: v is black.
  Then, v was dequeued before u.
  Claim 3: If v was dequeued before u, then $d[v] \leq d[u]$.

Proof of correctness

Proof cont...

- Case 3: v was gray.
  Vertex v was colored gray after dequeuing some vertex w earlier.
  So $d[v] = d[w] + 1$.
  By Claim 3, $d[w] \leq d[u]$ since w was dequeued before u.
  This gives: $d[v] = d[w] + 1 \leq d[u] + 1$.

Proof of correctness

Claim 2

Let $(u, v) \in E$. Then we have:

$$\delta(s, v) \leq \delta(s, u) + 1$$

Proof

If u is reachable from s, then:
Take the shortest path from s to u. Then take the edge (u, v).
This gives a path from s to v.
The shortest path from s to v can only be shorter than the above path.

$\square$

Proof of correctness

Claim 1

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

Proof

Induction on the number of enqueue operations.
Hypothesis: same as claim.
Base case: The time when the first vertex enqueued.
The first vertex enqueued is s. At this time we have:

- $\forall v \in V \setminus \{s\}, d[v] = \infty$
- $d[s] = \delta(s, s) = 0.$

Hence the claim holds for the base case.

Proof of correctness

Proof

Hypothesis: $\forall v \in V, d[v] \geq \delta(s, v)$
Step: A white (undiscovered) vertex v gets discovered while we are visiting a vertex u with $(u, v) \in E$.
From induction, we have: $d[u] \geq \delta(s, u)$.
The algorithm assigns $d[v] \leftarrow d[u] + 1$. So:

$$\begin{aligned} d[v] &= d[u] + 1 \\ &\geq \delta(s, u) + 1 \\ &\geq \delta(s, v) \end{aligned}$$

Last inequality follows from Claim 2.

$\square$

Proof of correctness

Claim 3

If v was dequeued before u, then $d[v] \leq d[u]$.

We will show a stronger claim:

Claim 4

If at some point, the queue contained $v_1, v_2, \ldots, v_r$ where v_1 was the head. Then:

(a) $d[v_1] \leq d[v_2] \leq \cdots \leq d[v_r]$

(b) $d[v_r] \leq d[v_1] + 1$

Proof of Claim 3:

Write down vertices in the order they went through the queue.

By claim 4 (a), the calculated d values for them are non-decreasing.

Vertex v will appear before u in this order.

Hence claim 3 follows. $\square$

Proof of correctness

Claim 4

If queue contains $v_1, v_2, \ldots, v_r$ where v_1 is the head. Then:

(a) $d[v_1] \leq d[v_2] \leq \cdots \leq d[v_r]$

(b) $d[v_r] \leq d[v_1] + 1$

Proof

Induction on number of queue operations.
Hypothesis: Same as claim. We show that the claim holds after every enqueue and dequeue.
Base case: The first queue operation - enqueuing s.
The claim trivially holds.

Proof of correctness

Claim 4

If queue contains $v_1, v_2, \ldots, v_r$ where v_1 is the head. Then:

(a) $d[v_1] \leq d[v_2] \leq \cdots \leq d[v_r]$

(b) $d[v_r] \leq d[v_1] + 1$

Proof

Step:

- **Dequeue:** After v_1 is dequeued, v_2 is the new head.
  Part (a): From induction,
  $d[v_1] \leq d[v_2] \leq d[v_3] \leq \cdots \leq d[v_r]$.
  Hence (a) holds.
  Part (b): From induction, $d[v_r] \leq d[v_1] + 1$. And so:

$$\begin{aligned} d[v_r] &\leq d[v_1] + 1 \\ &\leq d[v_2] + 1 \end{aligned}$$

Proof of correctness

Proof

- **Enqueue:** When a vertex v is enqueued:
  It was enqueued because:
  - it was undiscovered so far.
  - it was present in the adjacency list of a vertex u that was just dequeued.

  Since u was the previous head of the list, from induction we have:
  - $d[u] \leq d[v_1] \leq d[v_2] \leq \cdots \leq d[v_r]$.
  - $d[v_r] \leq d[u] + 1$.

  We assign $d[v] \leftarrow d[u] + 1$ and then enqueue v. Hence, we have:
  - $d[v_r] \leq d[u] + 1 = d[v]$
  - $d[v_1] \leq d[v_2] \leq \cdots \leq d[v_r] \leq d[v]$.

$\square$

Weighted Graphs

A weighted graph is a graph $G = (V, E)$ with a weight function:

$$w : E \to \mathbb{Z}$$

The weight of an edge $(u, v) \in E$ is $w((u, v))$.
For this lecture, we look at directed weighted graphs with weight function $w : E \to \mathbb{Z}^+$.

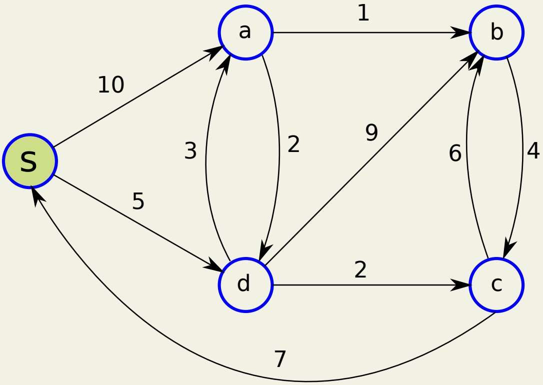
Shortest path in weighted graphs

Input:

- Graph $G = (V, E)$
- Weight function $w : E \to \mathbb{Z}^+$
- Source vertex $s \in V$.

Goal: Compute the shortest path from s to all reachable vertices.

Example graph



Dijkstra's Algorithm Pseudocode

Algorithm 2 Dijkstra's algorithm

For all $u \in V$, $d[u] \leftarrow \infty$, $\pi[u] \leftarrow$ NIL
$d[s] \leftarrow 0$
Initialize min-priority queue $Q \leftarrow V$
$S \leftarrow \emptyset$
while $Q \neq \emptyset$ **do**
    $u \leftarrow$ EXTRACT-MIN(Q)
    $S \leftarrow S \cup \{u\}$
    for each $v \in \mathcal{N}(u)$ **do**
        if $d[u] + w(u, v) < d[v]$ **then**
            $d[v] \leftarrow d[u] + w(u, v)$
            DECREASE-KEY$(v, d[v])$.
            $\pi[v] \leftarrow u$
        end if
    end for
end while

Proof of Correctness

Theorem

At the end of Dijkstra's algorithm, we have:

$$\forall u \in U, d[u] = \delta(s, u)$$

Proof

Loop Invariant:
At the start of each iteration, we have $\forall v \in S, d[v] = \delta(v)$.
Init: At the start of the first iteration, $S = \emptyset$. So the claim holds.
Maintenance: For the sake of contradiction, let u be the first vertex for which $d[u] \neq \delta(s, u)$.
Look at the start of the iteration where u was added to S.
If $u = s$ or u is not reachable from s, immediate contradiction.
So assume $u \neq s$ and u is reachable from s.

Proof of Correctness

Take a shortest path σ from s to u.
Let y be the first vertex on σ that is outside S.
Let $x \in S$ be the vertex on σ just before y.
So the path σ looks like $s \rightsquigarrow x \to y \rightsquigarrow u$.
Claim 1: $d[y] = \delta(s, y)$.
Since y appears before u in σ, we have $\delta(s, y) \le \delta(s, u)$.
Hence:

$$\begin{aligned} d[y] &= \delta(s, y) \\ &\le \delta(s, u) \\ &\le d[u] \end{aligned}$$

Although y and u were in $V \setminus S$, Extract-Min returned u.
This means $d[u] \le d[y]$. Hence:
$d[y] = \delta(s, y) = \delta(s, u) = d[u]$

$\square$

Claim 1

$d[y] = \delta(s, y)$

Proof

We chose u to be the first vertex for which $d[u] \neq \delta(s, u)$.
Since $x \in S$, $d[x] = \delta(s, x)$.
We updated $d[y]$ when we added x to S.
Now we note a *convergence* property:
If y is on a shortest path σ from s to u.
Then, the path formed by σ from s to y is a shortest path from s to y.

$\square$