Shortest Paths Problem

**Problem**: find shortest paths in a directed graph with edge lengths (e.g., Google maps)

Let’s Formalize the Problem

- Directed graph $G = (V, E)$ with nonnegative edge lengths $\ell(e) \geq 0$
- Define length of path $P$ consisting of edges $e_1, e_2, \ldots, e_k$ as
  $$\ell(P) = \ell(e_1) + \ell(e_2) + \ldots + \ell(e_k)$$
- Starting node $s$
- Let $d(v)$ be the length of shortest $s \rightarrow v$ path.
- **Problem**: Can we efficiently find $d(v)$ for all nodes $v \in V$?

**Question**: Why for all nodes at the same time?

Dijkstra Derivation

Suppose all edges have integer length. Can we use BFS to solve this problem?

Recall: nodes in layer $L_i$ are at distance $i$ from start.
If edge lengths are integers, and $C$ is the maximum length, the running time is (choose the most restrictive):

A. $O(C + m + n)$
B. $O(n + C \cdot m)$
C. $O(m + C \cdot n)$
D. $O(C \cdot (m + n))$

Hint: Why is BFS $O(m + n)$ and not $O(m)$?
## Dijkstra's Algorithm

### Induction Proof: Invariant (2)

Proof of Correctness

Running Time?

### Tracking the Shortest Path

Keep track of node that last updated arrival time $d'(v)$

Call it $\text{prev}(v) =$ predecessor in shortest path

### Proof of Correctness

Idea: nodes are explored in increasing order of distance from $s$.

At each step, we have:

$A$: set of nodes still to explore

$S = V \setminus A$: explored nodes, $d(v)$ assigned

Claim (invariant):

1. For $v \in S$, $d(v)$ is length of shortest $s \rightarrow v$ path
2. For $v \in A$, $d(v)$ is the length of the shortest $s \rightarrow v$ path with all nodes in $S$ except $v$.

Proof: By induction on $|S|$.

### Induction Proof

Base case: Initially $S = \emptyset$. Both (1) and (2) are true. ✓

Induction step:

- Assume the invariant is true for $|S| = k \geq 0$.
- Let $v = \text{next node added to } S$, so we set $d(v) = d'(v)$
- We claim (1) is true: the shortest $s \rightarrow v$ path has length $d(v) = d'(v)$.
- By (2), $d'(v)$ is length of the shortest $s \rightarrow v$ path with all prior nodes in $S$. We claim this is the shortest $s \rightarrow v$ path overall.
- Why? Every path to $v$ must leave $S$ eventually; if it first hops to some other node $y$ outside of $S$, it is already at least as long as $d'(v)$, because $d'(v) \leq d'(y)$.
- Paths can’t get shorter after leaving $S$ because edge lengths are non-negative.

### Induction Proof: Invariant (2)

Invariant (2) is directly maintained by the algorithm: after adding $v$ to $S$, it updates $d'(w)$ for all neighbors $w$ if a shorter path is found through $v$.

for all edges $(v, w)$ where $w \in A$ do

if $d(v) + \ell(v, w) < d'(w)$ then

$d'(w) = d(v) + \ell(v, w)$
Dijkstra's algorithm works for nonnegative edges. (We'll discuss the Bellman-Ford algorithm, which can handle negative edges.)

In general, there exists a shortest $s \rightarrow v$ path if

A. There are no negative-length edges on any path $s \rightarrow v$

B. There is no negative-length cycle on any path $s \rightarrow v$

C. Any path $s \rightarrow v$ that has a negative-length cycle also has a positive-length cycle

D. Any path $s \rightarrow v$ that has a negative-length cycle also has a positive-length cycle, longer in absolute value