Agenda
1. BFS as a shortest path algo.
2. Dijkstra
3. Which road to add?

BFS (Breadth-First Search)

- Defn - BFS on an unweighted graph \( G = (V,E) \) starting at \( s \in V \), assigns each node \( v \in V \) a level \( (v) = \# \) of edges in shortest path from \( s \).

Example:
6-degrees of Kevin Bacon
Consider the friend graph of the world
- \( V = \) every person ever to live in the world
- \( E = \) \( (u,v) \) if \( u \) and \( v \) are friends (or know each other)

Say you are node \( x \in V \). How far are you from Kevin Bacon? Use BFS! What level is KB on in \( BFS(G,v) \)?
2] Dijkstra

- What about weighted graphs?

- Dijkstra finds the following:

  Input: $G = (V, E)$; constant node $s$; weights $w_e \geq 0 \forall e \in E$.

  Output: Shortest Path Tree rooted at $s$ to all $v \in V$.

Ex: SPT of $G$:

$G = s \quad 10 \quad 9 \quad d$

$\quad 4 \quad 1 \quad 2 \quad b \quad 20$

$\quad 2 \quad c$

SPT$(s)$:

Note: Bellman-Ford finds SPs when $w_e$ is arbitrary but no negative cycles in $G$. Why no neg. cycles?

- Dijkstra uses a priority queue + BFS-like search.

3] Which road to add? (4, 20)

Input: $G = (V, E)$ cities & roads; $w_e \geq 0$ length of road $e \in E$.

Output: Find $e' \in E'$ whose addition to $G$ would result in the largest decrease in SP distance between $s$ and $t$.

Idea: Learn $SPT(G, s)$ and $SPT(G', t)$.

For $e' = (u, v) \in E'$:

- Compute $d_S(u) + f_t(u) + f_t(v) + d_t(v)$.
- Find smallest.
Alg. \text{Road}(G, l_e, E', l_e', s, t)

1) Run \text{Dijkstra}(G, l_e, s) and \text{Dijkstra}(G, l_e, t) to get $d_s(v)$ and $d_t(v)$ (as defined earlier).

2) $\text{min-sp} = \min(d_s(t), d_t(s))$ and $\text{ret-edge} = \emptyset$.

3) For each $e' = (u, v) \in E'$:
   
   $\text{if}(d_s(u) + l_e + d_t(v) < \text{min-sp})$ then $\text{min-sp} = d_s(u) + l_e + d_t(v)$ and $\text{ret-edge} = e'$.

   $\text{if}(d_s(v) + l_e' + d_t(u) < \text{min-sp})$ then $\text{min-sp} = d_s(v) + l_e' + d_t(u)$ and $\text{ret-edge} = e'$.

4) return $\text{ret-edge}$.

$RT = O((|V| + |E|) \log |V|)$ for Dijkstra with binary heap.

For loop is $O(|E'|)$ so $O((|V| + |E|) \log |V| + |E'|)$

Correctness Claim: \text{Road}() returns $e'$ iff $e'$ is the road whose addition minimizes the SP dist from $s$ to $t$.

$\Rightarrow$ Assume \text{Road}() returns $e' = (u, v)$. Then either $d_s(u) + l_e + d_t(v)$ or $d_s(v) + l_e' + d_t(u)$ is smallest of all $E'$, both of which are paths from $s$ to $t$.

So, it is the new SP for $T$. They are also both SPs since $d_s$ is a SP and $d_t$ is also a SP.

$\Leftarrow$ Let $e' = (u, v)$ be the road whose addition minimizes SP dist between $s$ and $t$. Then either $d_s(u) + l_e + d_t(v)$ or $d_s(v) + l_e' + d_t(u)$ must be the length of the shortest path between $s$ and $t$. This is what \text{Road}() tests for and minimizes.