Dijkstra's SSSP Algorithm

Single Source Shortest Paths (SSSP)

“Extension” of BFS exploration to weighted graph

Input: a directed graph $G = (V, E)$ with non-negative weights $w : E \to \mathbb{R} \geq 0$

and a source vertex $S \in V$.

Goal: find a shortest path from s to any $v \in V$ in terms of total weight.

Remark: even if we just care about shortest path between s and another vertex t, running time is asymptotically the same.

Remark: One can also consider negative weights, and there are algorithms for this extension, like Bellmar-Ford, running in time $O(V \cdot E)$.

Remark: the case of all weights being 1 (i.e. unweighted) is solved by BFS.

DIJKSTRA(G, w, s):

for each $v \in V$:

    $v.d = \infty$

    $v.\pi = \mathrm{NIL}$

$s.d = 0$

$Q = \mathrm{PRIORITYQUEUE}(V)$

while $Q \neq 0$:

    $u = \mathrm{EXTRACTMIN}(Q)$

    for each $v \in \mathrm{Adj}[u]$:

        if $v.d > u.d + w(u, v)$:

            $\mathrm{DECREASEKEY}(Q, v, u.d + w(u, v))$

            $v.\pi = u$

Runtime:

Initialization + $|V| \times$EXTRACTMIN + $|E| \times$DECREASEKEY

So, just like in Prim, using binary heap, total runtime is

$$O(V + VlogV + ElogV) = O((V + E)logV).$$

With Fibonacci heaps, $O(V + VlogV + E) = O(VlogV + E)$.

Correctness:

Let $\delta(u, v)$ be weight of the shortest path from u to v.

Theorem: Dijkstra’s algorithm terninates with $u.d = \delta(s, u)$ for all $u \in V$.

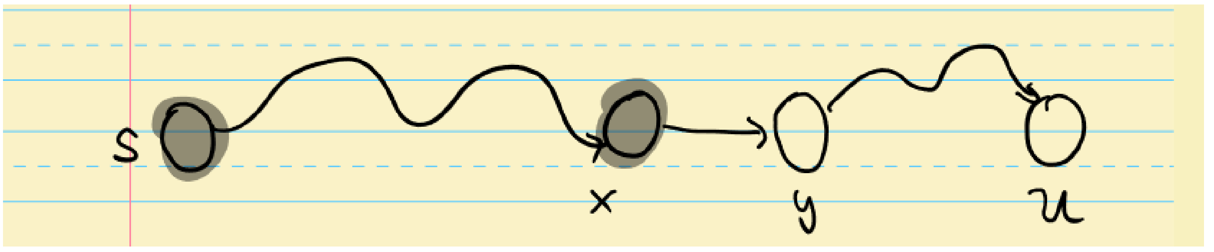
Loop invariant: each vertex v that is no longer in Q satisfies $v.d = \delta(s, v)$.

Initialization: trivial.

Maintenance: assume the invariant holds for all black vertices and that we're about to add u.

Consider a shortest path from s to u.

Let x and y be two consecutive vertices along the path such that x is black and y is white.



By loop invariant, $x.d = \delta(s, x)$.

Because x explore all its neighbors, $y.d \leq x.d + w(x, y) = \delta(s, x) + w(x, y) = \delta(s, y)$

Since $y.d$ never goes below $\delta(s, y)$, we have $y.d = \delta(s, y)$.

Finally, because u is chosen from the priority queue, $u.d \leq y.d = \delta(s, y) \leq \delta(s, u)$,

and therefore $u.d \leq y.d = \delta(s, u)$.