Shortest Path Algorithms

Hui Chen [a]

[a]CUNY Brooklyn College

November 4, 2020

Dijkstra's Algorithm

Algorithm 1: Dijkstra's Shortest Path Algorithm

```
Input    : G, Graph G = (V_G, E_G)
Input    : s, s ∈ V_G is the source node of the shortested paths in Graph G
Function DijkstraShortestPath(G,s)
    for n ∈ V_G do
        d(s, n) ← ∞
        p(n) ← nil
        N_G = V_G
    d(s, s) ← 0
    while N_G ≠ φ do
        u ← u | d(s, u) = min_{∀n∈N_G} d(s, n)
        N_G ← N_G \ {u}
        for v ∈ Neighbor(G, u) do
            l ← d(s, u) + w(u, v)
            if l < d(s, v) then
                d(s, v) ← l
                p(v) ← u
    return d,p
```

Bellman-Ford Algorithm

Algorithm 2: Bellman-Ford Shortest Path Algorithm

Input : G, Graph $G = (V_G, E_G, W_G)$
Input : s, $s \in V_G$ is the source node of the shortested paths in Graph G
Function `BellmanFordShortestPath`(G,s)

```
for n ∈ V_G do
    d(s, n) ← ∞
    p(n) ← nil
d(s, s) ← 0
for u ∈ V_G do
    for (u, v) ∈ E_G do
        l ← d(s, u) + w(u, v)
        if l < d(s, v) then
            d(s, v) ← l
            p(v) ← u
for (u, v) ∈ E_G do
    if d(s, v) > d(s, u) + w(u, v) then
        RaiseError("Negative Cycle")
return d,p
```