

Shortest Path Algorithms

Hui Chen ^a

^aCUNY Brooklyn College

November 2, 2023

Dijkstra's Algorithm

Algorithm 1: Dijkstra's Shortest Path Algorithm

Input : G , Graph $G = (V_G, E_G)$

Input : $s, s \in V_G$ is the source node of the shortest paths in Graph G

Function DijkstraShortestPath(G, s)

```
    for  $n \in V_G$  do
         $d(s, n) \leftarrow \infty$ 
         $p(n) \leftarrow nil$ 
     $N_G = V_G$ 
     $d(s, s) \leftarrow 0$ 
    while  $N_G \neq \emptyset$  do
         $u \leftarrow u | d(s, u) = \min_{\forall n \in N_G} d(s, n)$ 
         $N_G \leftarrow N_G \setminus \{u\}$ 
        for  $v \in \text{Neighbor}(G, u)$  do
             $l \leftarrow d(s, u) + w(u, v)$ 
            if  $l < d(s, v)$  then
                 $d(s, v) \leftarrow l$ 
                 $p(v) \leftarrow u$ 
    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 shortest paths in Graph G

Function BellmanFordShortestPath(G, s)

```
    for  $n \in V_G$  do
         $d(s, n) \leftarrow \infty$ 
         $p(n) \leftarrow nil$ 
     $d(s, s) \leftarrow 0$ 
    for  $u \in V_G$  do
        for  $(u, v) \in E_G$  do
             $l \leftarrow d(s, u) + w(u, v)$ 
            if  $l < d(s, v)$  then
                 $d(s, v) \leftarrow l$ 
                 $p(v) \leftarrow u$ 

    for  $(u, v) \in E_G$  do
        if  $d(s, v) > d(s, u) + w(u, v)$  then
            RaiseError("Negative Cycle")

    return  $d, p$ 
```
