Data Structures And Algorithms

Office : LWSN 3151-1
Office Hours : Tuesdays 10-12pm
Topics

- Single Source Shortest Path
  - Bellman Ford Algorithm
- Strongly Connected Components
Single Source Shortest Path

- Optimal substructure
- Edge relaxation
- Negative-weight edges
Optimal substructure

- let \( p = s \ldots v \ldots d \) be a shortest path from vertex \( s \) to vertex \( d \)
- Then subpath \( s \ldots v \) and \( v \ldots d \) are both shortest paths as well
Edge relaxation

- To relax an edge \((v,w)\) means to test whether the best known way from \(s\) to \(w\) is to go from \(s\) to \(v\), then take the edge from \(v\) to \(w\).
- If so, update our data structures.
Edge relaxation

- **v->w is ineligible**: Black edges are in $\text{edgeTo}[]$.

- **v->w is eligible**: Weight of v->w is 1.3. 

- **no changes**: No changes occur.

- **edgeTo[w]**: `edgeTo[w]` is updated.

- **no longer in SPT**: Node is no longer in the Shortest Path Tree (SPT).
Edge relaxation

- RELAX (u, v, w)
  - if (v.d > u.d + w(u, v))
    - v.d = u.d + w(u, v)
    - v.p = u
Negative-weight edges

- There can be negative-weight edges
- But no negative-weight cycles
  - The concept of a shortest path is meaningless if there is a negative cycle.
Negative-weight edges

tinyEWDnc.txt

\[ V \rightarrow 8 \]
\[ 15 \rightarrow E \]

\begin{align*}
4 & 5 & 0.35 \\
5 & 4 & -0.66 \\
4 & 7 & 0.37 \\
5 & 7 & 0.28 \\
7 & 5 & 0.28 \\
5 & 1 & 0.32 \\
0 & 4 & 0.38 \\
0 & 2 & 0.26 \\
7 & 3 & 0.39 \\
1 & 3 & 0.29 \\
2 & 7 & 0.34 \\
6 & 2 & 0.40 \\
3 & 6 & 0.52 \\
6 & 0 & 0.58 \\
6 & 4 & 0.93 \\
\end{align*}

shortest path from 0 to 6

\[ 0 \rightarrow 4 \rightarrow 7 \rightarrow 5 \rightarrow 4 \rightarrow 7 \rightarrow 5 \ldots \rightarrow 1 \rightarrow 3 \rightarrow 6 \]
Topics

- Single Source Shortest Path
  - Bellman Ford Algorithm
- Strongly Connected Components
Bellman Ford Algorithm

- INITIALIZE-SINGLE-SOURCE(G, s)
  - for each vertex v in G.V
    - v.d = infinity
    - v.p = NIL
  - s.d = 0
Bellman Ford Algorithm

• BELLMAN-FORD(G, w, s)
  ▪ INITIALIZE-SINGLE-SOURCE(G, s)
  ▪ for i = 1 to |G:V| - 1
    ▪ for each edge (u, v) in G.E
      ▪ RELAX(u, v, w)
    ▪ for each edge (u, v) in G:E
      ▪ if (v.d > u.d + w(u, v))
        ▪ return FALSE
  ▪ return TRUE
Topics

- Single Source Shortest Path
  - Bellman Ford Algorithm
- Strongly Connected Components
Strongly Connected Components

- two vertices of directed graph are in the same component \textbf{iff} they are reachable from each other.

\{a, b, e\}, \{c, d\}, \{f, g\}, and \{h\}
Strongly Connected Components

- $G'$ is $G$ with all edges reversed

- **Observation**: $G$ & $G'$ have the same SCC
Strongly Connected Components

- STRONGLY-CONNECTED-COMPONENTS (G)
  - Call DFS(G) to compute finishing times f[u] for all u.
  - Compute G'
  - Call DFS(GT), but in the main loop, consider vertices in order of decreasing finishing times
  - Output the vertices in each tree of the depth-first forest formed in second DFS as a separate SCC.
Strongly Connected Components

- STEP 1
Strongly Connected Components

- **STEP 1**

- **STEP 2**
Strongly Connected Components

- **STEP 3**

- **STEP 2**
Questions