COL106: Data Structures and Algorithms

Ragesh Jaiswal, IIT Delhi

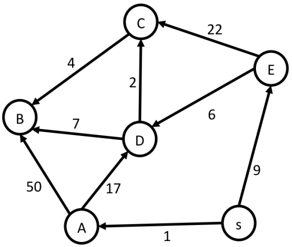
Greedy Algorithms: Single Source Shortest Path

- Path length: Let $G = (V, E)$ be a weighted directed graph. Given a path in G, the length of a path is defined to be the sum of lengths of the edges in the path.
- Shortest path: The shortest path from u to v is the path with minimum length.

- Path length: Let $G = (V, E)$ be a weighted directed graph. Given a path in G, the length of a path is defined to be the sum of lengths of the edges in the path.
- Shortest path: The shortest path from u to v is the path with minimum length.

Problem

Single source shortest path: Given a weighted, directed graph $G = (V, E)$ with positive edge weights and a source vertex s, find the shortest path from s to all other vertices in the graph.



Problem

Single source shortest path: Given a weighted, directed graph $G = (V, E)$ with positive edge weights and a source vertex s, find the shortest path from s to all other vertices in the graph.

- Claim 1: Shortest path is a *simple* path.

Problem

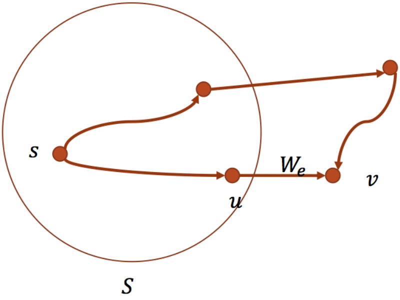
Single source shortest path: Given a weighted, directed graph $G = (V, E)$ with positive edge weights and a source vertex s, find the shortest path from s to all other vertices in the graph.

- Claim 1: Shortest path is a *simple* path.
- Claim 2: For any vertex $x \in V$, let $l(s, x)$ denote the length of the shortest path from s to vertex x. Let S be any subset of vertices containing s. Let $e = (u, v)$ be an edge such that:
  1. $u \in S$, $v \in V \setminus S$ *(that is, (u, v) is a cut edge)*,
  2. $(l(s, u) + W_e)$ is the least among all such cut edges.

  Then $l(s, v) = l(s, u) + W_e$.

- Claim 2: For any vertex $x \in V$, let $l(s, x)$ denote the length of the shortest path from s to vertex x. Let S be any subset of vertices containing s. Let $e = (u, v)$ be an edge such that:
  1. $u \in S$, $v \in V \setminus S$ *(that is, (u, v) is a cut edge)*,
  2. $(l(s, u) + W_e)$ is the least among all such cut edges.

  Then $l(s, v) = l(s, u) + W_e$.



- Claim 2: For any vertex $x \in V$, let $l(s, x)$ denote the length of the shortest path from s to vertex x. Let S be any subset of vertices containing s. Let $e = (u, v)$ be an edge such that:
  1. $u \in S$, $v \in V \setminus S$ *(that is, (u, v) is a cut edge)*,
  2. $(l(s, u) + W_e)$ is the least among all such cut edges.

  Then $l(s, v) = l(s, u) + W_e$.

Algorithm

`Dijkstra's Algorithm`(G, s)

- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

- Claim 2: For any vertex $x \in V$, let $l(s, x)$ denote the length of the shortest path from s to vertex x. Let S be any subset of vertices containing s. Let $e = (u, v)$ be an edge such that:
  1. $u \in S$, $v \in V \setminus S$ *(that is, (u, v) is a cut edge)*,
  2. $(l(s, u) + W_e)$ is the least among all such cut edges.

  Then $l(s, v) = l(s, u) + W_e$.

Algorithm

Dijkstra's Algorithm(G, s)
- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

- What is the running time of the above algorithm?

- Claim 2: For any vertex $x \in V$, let $l(s,x)$ denote the length of the shortest path from s to vertex x. Let S be any subset of vertices containing s. Let $e = (u,v)$ be an edge such that:
  1. $u \in S$, $v \in V \setminus S$ *(that is, (u,v) is a cut edge)*,
  2. $(l(s,u) + W_e)$ is the least among all such cut edges.

  Then $l(s,v) = l(s,u) + W_e$.

Algorithm

Dijkstra's Algorithm(G,s)
- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u,v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

- What is the running time of the above algorithm?
  - Same as that of the Prim's algorithm. $O(|E| \cdot \log|V|)$.

Algorithm

Dijkstra's Algorithm(G, s)
- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

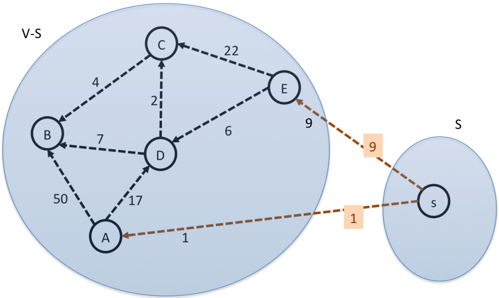


Figure: $d(s) = 0$

Algorithm

`Dijkstra's Algorithm`(G, s)
- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

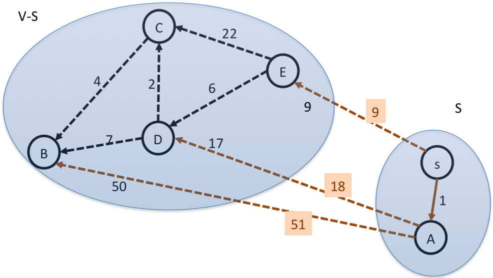


Figure: $d(s) = 0; d(A) = 1$

Algorithm

Dijkstra's Algorithm(G, s)

- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

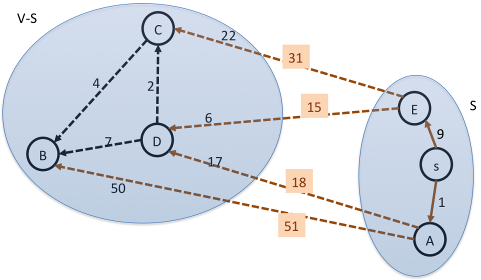


Figure: $d(s) = 0; d(A) = 1; d(E) = 9$

Algorithm

`Dijkstra's Algorithm`(G, s)
- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

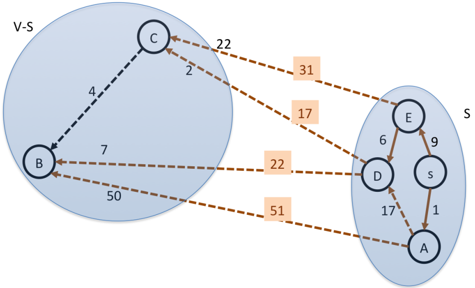


Figure: $d(s) = 0; d(A) = 1; d(E) = 9; d(D) = 15$

Algorithm

`Dijkstra's Algorithm`(G, s)
- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

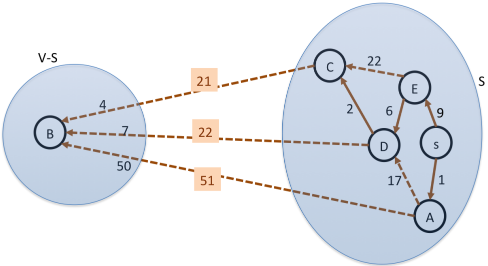


Figure: $d(s) = 0; d(A) = 1; d(E) = 9; d(D) = 15; d(C) = 17$

Algorithm

`Dijkstra's Algorithm`(G, s)

- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

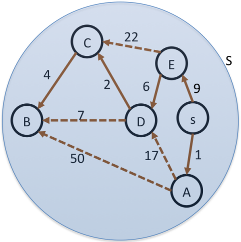


Figure: $d(s) = 0; d(A) = 1; d(E) = 9; d(D) = 15; d(C) = 17; d(B) = 21$

Algorithm

Dijkstra's Algorithm(G, s)

- $S \leftarrow \{s\}$
- $d(s) \leftarrow 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge across $(S, V \setminus S)$ with minimum value of $d(u) + W_e$
  - $d(v) \leftarrow d(u) + W_e$
  - $S \leftarrow S \cup \{v\}$

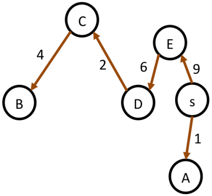


Figure: The algorithm also implicitly produces a *shortest path tree* that gives the shortest paths from s to all vertices.

End