Dynamic Programming: Shortest Paths and DFA to Reg Expressions

Lecture 18

Thursday, October 29, 2020

LaTeXed: October 7, 2020 12:53

18.1
Shortest Paths with Negative Length Edges

18.1.1
Why Dijkstra's algorithm fails with negative edges

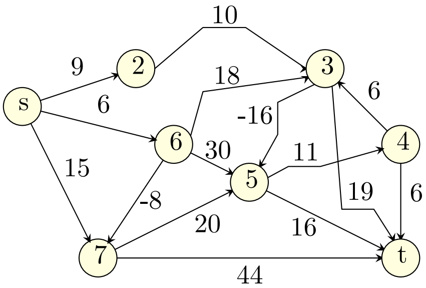
Single-Source Shortest Paths with Negative Edge Lengths

Problem statement

Single-Source Shortest Path Problems

Input: A directed graph $G = (V, E)$ with arbitrary (including negative) edge lengths. For edge $e = (u, v)$, $\ell(e) = \ell(u, v)$ is its length.

1. Given nodes s, t find shortest path from s to t.
2. Given node s find shortest path from s to all other nodes.



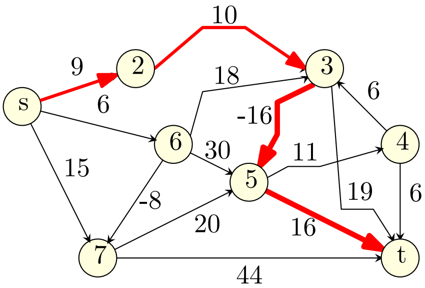
Single-Source Shortest Paths with Negative Edge Lengths

Problem statement

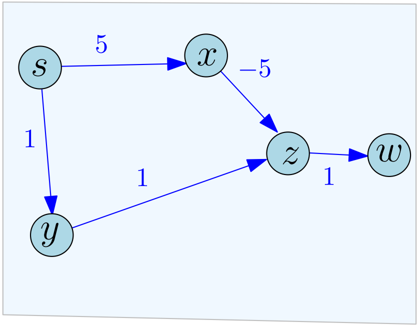
Single-Source Shortest Path Problems

Input: A directed graph $G = (V, E)$ with arbitrary (including negative) edge lengths. For edge $e = (u, v)$, $\ell(e) = \ell(u, v)$ is its length.

1. Given nodes s, t find shortest path from s to t.
2. Given node s find shortest path from s to all other nodes.



What are the distances computed by Dijkstra's algorithm?

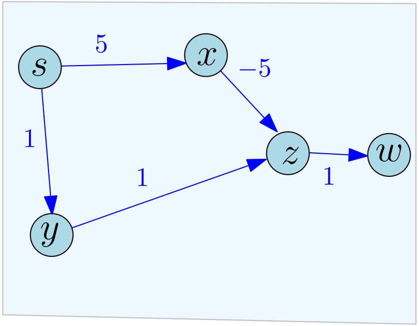


The distance as computed by Dijkstra algorithm starting from s:

- (A) $s = 0$, $x = 5$, $y = 1$, $z = 0$.
- (B) $s = 0$, $x = 1$, $y = 2$, $z = 5$.
- (C) $s = 0$, $x = 5$, $y = 1$, $z = 2$.
- (D) IDK.

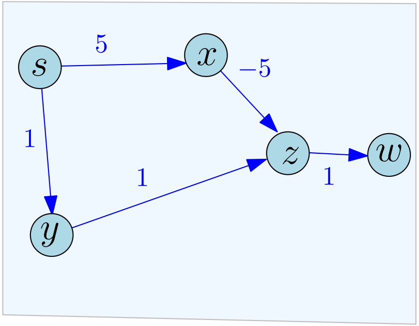
Dijkstra's Algorithm and Negative Lengths

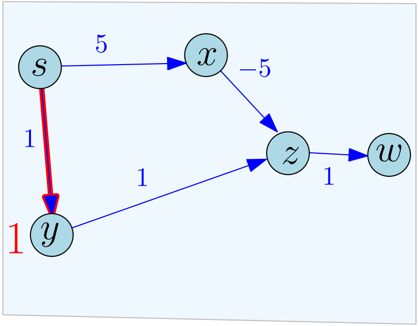
With negative length edges, Dijkstra's algorithm can fail



Dijkstra's Algorithm and Negative Lengths

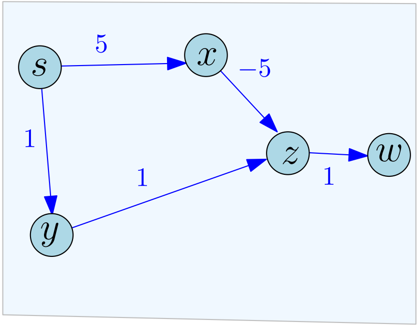
With negative length edges, Dijkstra's algorithm can fail

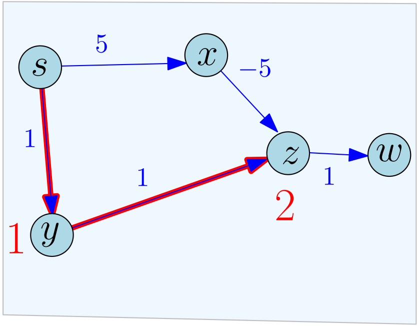




Dijkstra's Algorithm and Negative Lengths

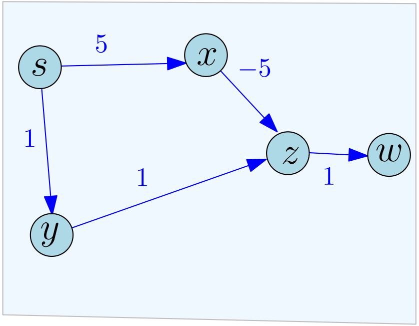
With negative length edges, Dijkstra's algorithm can fail

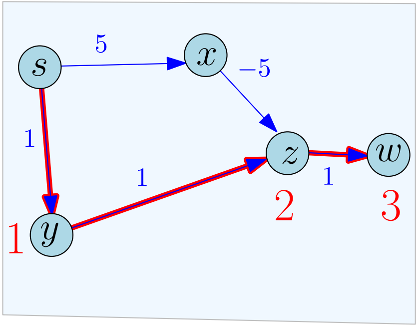




Dijkstra's Algorithm and Negative Lengths

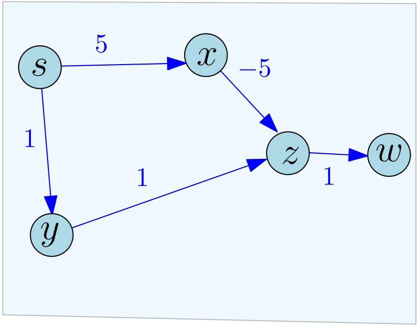
With negative length edges, Dijkstra's algorithm can fail

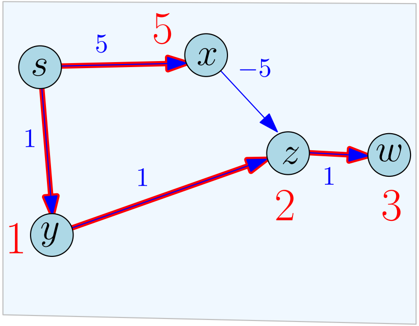




Dijkstra's Algorithm and Negative Lengths

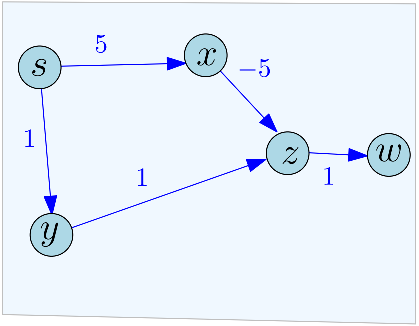
With negative length edges, Dijkstra's algorithm can fail

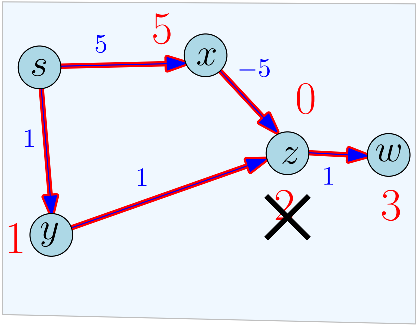




Dijkstra's Algorithm and Negative Lengths

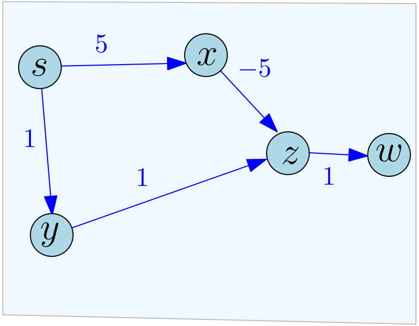
With negative length edges, Dijkstra's algorithm can fail

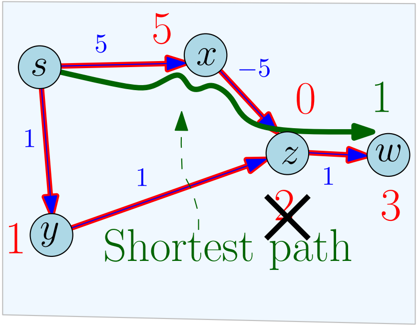




Dijkstra's Algorithm and Negative Lengths

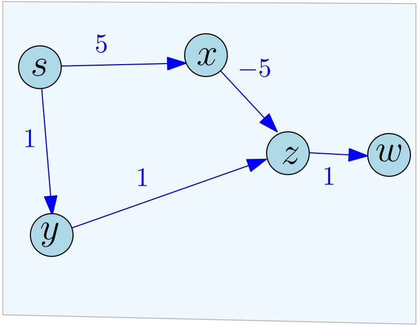
With negative length edges, Dijkstra's algorithm can fail

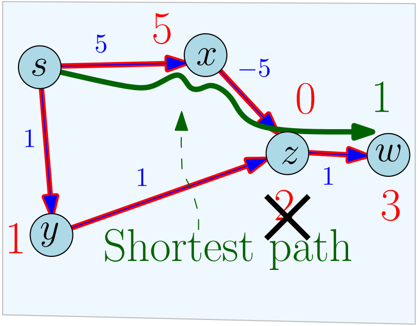




Dijkstra's Algorithm and Negative Lengths

With negative length edges, Dijkstra's algorithm can fail





False assumption: Dijkstra's algorithm is based on the assumption that if $s = v_0 \to v_1 \to v_2 \ldots \to v_k$ is a shortest path from s to v_k then $dist(s, v_i) \leq dist(s, v_{i+1})$ for $0 \leq i < k$. Holds true only for non-negative edge lengths.

Shortest Paths with Negative Lengths

Lemma 18.1.

Let G be a directed graph with arbitrary edge lengths. If $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k$ is a shortest path from s to v_k then for $1 \leq i < k$:

1. $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_i$ *is a shortest path from* s *to* v_i
2. *False:* $\mathrm{dist}(s, v_i) \leq \mathrm{dist}(s, v_k)$ *for* $1 \leq i < k$*. Holds true only for non-negative edge lengths.*

Cannot explore nodes in increasing order of distance! We need other strategies.

Shortest Paths with Negative Lengths

Lemma 18.1.

Let G be a directed graph with arbitrary edge lengths. If $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k$ is a shortest path from s to v_k then for $1 \leq i < k$:

1. *$s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_i$ is a shortest path from s to v_i*
2. *False: $\textbf{dist}(s, v_i) \leq \textbf{dist}(s, v_k)$ for $1 \leq i < k$. Holds true only for non-negative edge lengths.*

Cannot explore nodes in increasing order of distance! We need other strategies.

Shortest Paths with Negative Lengths

Lemma 18.1.

Let G be a directed graph with arbitrary edge lengths. If $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k$ is a shortest path from s to v_k then for $1 \leq i < k$:

1. $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_i$ *is a shortest path from* s *to* v_i
2. *False:* $\textbf{dist}(s, v_i) \leq \textbf{dist}(s, v_k)$ *for* $1 \leq i < k$. *Holds true only for non-negative edge lengths.*

Cannot explore nodes in increasing order of distance! We need other strategies.

THE END

...

(for now)