

COL 351 : ANALYSIS & DESIGN OF ALGORITHMS

LECTURE 24

DYNAMIC PROGRAMMING I :

BELLMAN FORD ALGORITHM AND ALL PAIRS SHORTEST PATHS

OCT 07, 2026

|

ROHIT VAISH

REVISITING SINGLE SOURCE SHORTEST PATHS

SINGLE - SOURCE SHORTEST PATH PROBLEM

input: A directed graph $G = (V, E)$, starting vertex s ,
a non negative length l_e for each edge $e \in E$

output: $\text{dist}(s, v)$ for every vertex $v \in V$.

SINGLE - SOURCE SHORTEST PATH PROBLEM

input: A directed graph $G = (V, E)$, starting vertex s ,
a non negative length l_e for each edge $e \in E$

output: $\text{dist}(s, v)$ for every vertex $v \in V$.

length of the shortest path from s to v if $s \rightsquigarrow v$ path exists
 ∞ otherwise

ON DIJKSTRA'S ALGORITHM

ON DIJKSTRA'S ALGORITHM



$O(m \log n)$ running time with heaps

ON DIJKSTRA'S ALGORITHM



$O(m \log n)$ running time with heaps



not always correct with negative edge lengths
(e.g., if edges $\leftrightarrow$ financial transactions)

ON DIJKSTRA'S ALGORITHM



$O(m \log n)$ running time with heaps



not always correct with negative edge lengths
(e.g., if edges $\leftrightarrow$ financial transactions)

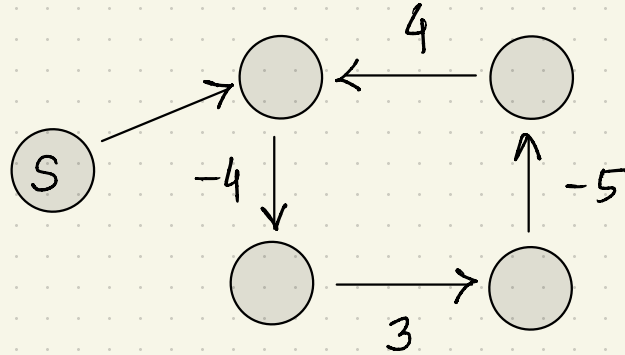
Solution : Bellman-Ford algorithm

NEGATIVE CYCLES

How to define shortest paths in the presence of negative cycles?

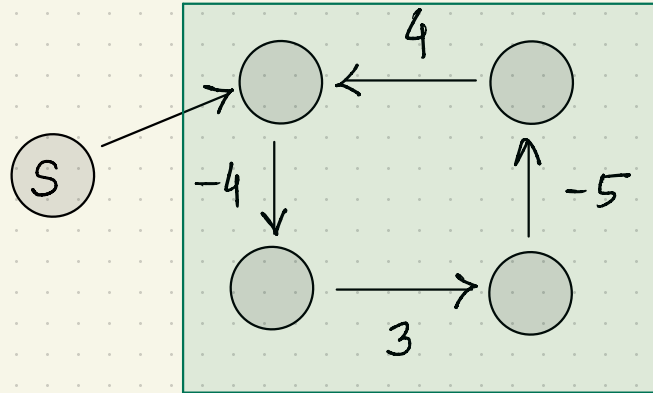
NEGATIVE CYCLES

How to define shortest paths in the presence of negative cycles?



NEGATIVE CYCLES

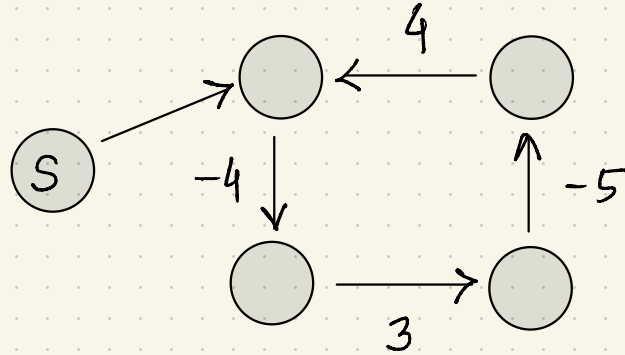
How to define shortest paths in the presence of negative cycles?



cycle with weight -2

NEGATIVE CYCLES

How to define shortest paths in the presence of negative cycles?

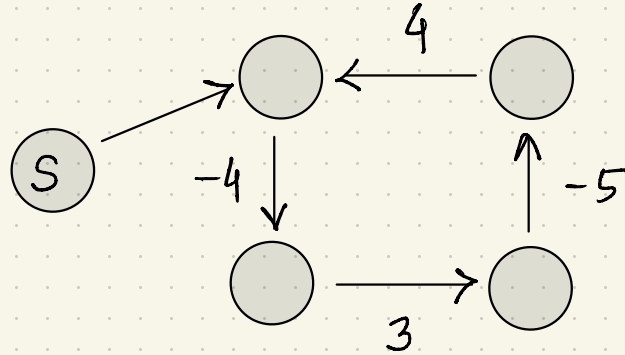


If we **include** cycles in shortest paths

then length of the shortest path is **undefined** (or $-\infty$).

NEGATIVE CYCLES

How to define shortest paths in the presence of negative cycles?

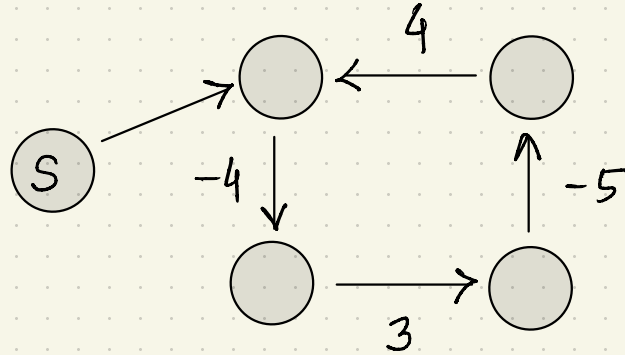


If we require shortest **cycle-free** path

then the problem is **NP-complete** (no known poly-time algo.)

NEGATIVE CYCLES

How to define shortest paths in the presence of negative cycles?



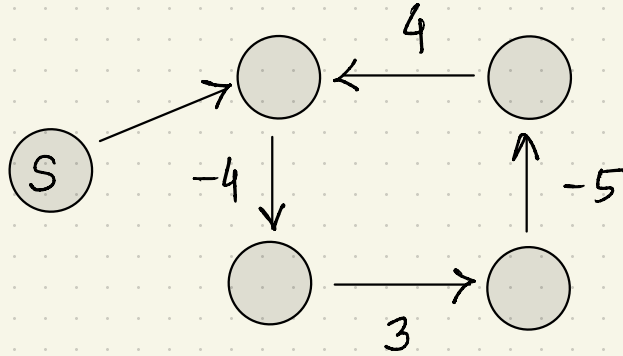
If we require shortest **cycle-free** path

then the problem is **NP-complete** (no known poly-time algo.)

— encodes "longest path" problem

NEGATIVE CYCLES

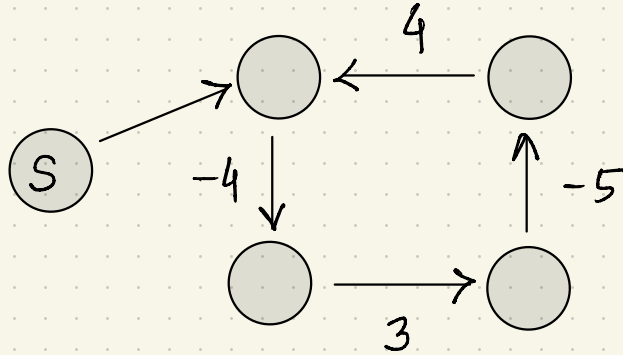
How to define shortest paths in the presence of negative cycles?



Solution: Allow negative length edges but not negative cycles.

NEGATIVE CYCLES

How to define shortest paths in the presence of negative cycles?



Solution: Allow negative length edges but not negative cycles.

will show how to quickly check this

SINGLE - SOURCE SHORTEST PATH PROBLEM

SINGLE - SOURCE SHORTEST PATH PROBLEM

input: A directed graph $G = (V, E)$, starting vertex s ,
a possibly negative length l_e for each edge $e \in E$

SINGLE - SOURCE SHORTEST PATH PROBLEM

input: A directed graph $G = (V, E)$, starting vertex s ,
a possibly negative length l_e for each edge $e \in E$ possibly contains negative cycles

SINGLE - SOURCE SHORTEST PATH PROBLEM

input: A directed graph $G = (V, E)$, starting vertex s ,
a possibly negative length l_e for each edge $e \in E$

output:

SINGLE - SOURCE SHORTEST PATH PROBLEM

input: A directed graph $G = (V, E)$, starting vertex s ,
a possibly negative length l_e for each edge $e \in E$

output: ① for all destinations $v \in V$, compute the length of the
shortest $s-v$ path, or
② output a negative cycle.

OPTIMAL SUBSTRUCTURE

OPTIMAL SUBSTRUCTURE

Lemma: Let $G = (V, E)$ be a directed graph with edge lengths l_e and starting vertex s .

OPTIMAL SUBSTRUCTURE

Lemma: let $G = (V, E)$ be a directed graph with edge lengths l_e and starting vertex s . may or may not have a negative cycle

OPTIMAL SUBSTRUCTURE

Lemma: Let $G = (V, E)$ be a directed graph with edge lengths l_e and starting vertex s .

For every destination $v \in V$ and every $i \in \{1, 2, \dots\}$, let

P be the shortest $s-v$ path with at most i edges.



OPTIMAL SUBSTRUCTURE

Lemma: Let $G = (V, E)$ be a directed graph with edge lengths l_e and starting vertex s .

For every destination $v \in V$ and every $i \in \{1, 2, \dots\}$, let

P be the shortest $s-v$ path with at most i edges.

Case I: P has $(i-1)$ or fewer edges

$\Rightarrow P$ is the shortest $s-v$ path with at most $(i-1)$ edges.

OPTIMAL SUBSTRUCTURE

Lemma: Let $G = (V, E)$ be a directed graph with edge lengths l_e and starting vertex s .

For every destination $v \in V$ and every $i \in \{1, 2, \dots\}$, let

P be the shortest $s-v$ path with at most i edges.

Case I: P has $(i-1)$ or fewer edges

$\Rightarrow P$ is the shortest $s-v$ path with at most $(i-1)$ edges.

Case II: P has i edges

OPTIMAL SUBSTRUCTURE

Lemma: Let $G = (V, E)$ be a directed graph with edge lengths l_e and starting vertex s .

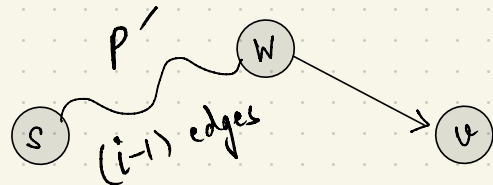
For every destination $v \in V$ and every $i \in \{1, 2, \dots\}$, let

P be the shortest $s-v$ path with at most i edges.

Case I: P has $(i-1)$ or fewer edges

$\Rightarrow P$ is the shortest $s-v$ path with at most $(i-1)$ edges.

Case II: P has i edges



OPTIMAL SUBSTRUCTURE

Lemma: Let $G = (V, E)$ be a directed graph with edge lengths l_e and starting vertex s .

For every destination $v \in V$ and every $i \in \{1, 2, \dots\}$, let

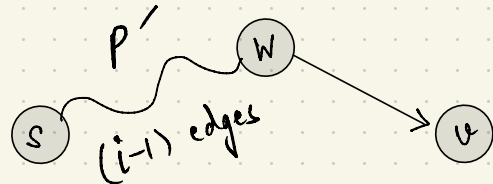
P be the shortest $s-v$ path with at most i edges.

Case I: P has $(i-1)$ or fewer edges

$\Rightarrow P$ is the shortest $s-v$ path with at most $(i-1)$ edges.

Case II: P has i edges

$\Rightarrow P'$ is the shortest $s-w$ path with at most $(i-1)$ edges



THE RECURRENCE

THE RECURRENCE

Define $L_{i,v} :=$ length of shortest $s-v$ path with $\leq i$ edges

THE RECURRENCE

Define $L_{i,v} :=$ length of shortest $s-v$ path with $\leq i$ edges

with cycles allowed

defined as $+\infty$ if no $s-v$ path with $\leq i$ edges

THE RECURRENCE

Define $L_{i,v} :=$ length of shortest $s-v$ path with $\leq i$ edges

For every $v \in V$ and every $i \in \{0, 1, 2, \dots\}$

$$L_{i,v} = \min \begin{cases} L_{(i-1),v} & // \text{Case I} \\ \min_{(w,v) \in E} L_{(i-1),w} + l_{w,v} & // \text{Case II} \end{cases}$$

BELLMAN-FORD ALGORITHM

BELLMAN-FORD ALGORITHM

$A := n \times n$ dimensional array

// budget $\times$ destinations

// base cases

$A[0, s] := 0$, $A[0, v] := +\infty$ for all $v \neq s$

// recurrence

for $i \in \{1, 2, \dots, n-1\}$

for each $v \in V$

$$A[i, v] = \min \begin{cases} A[i-1, v] \\ \min_{(w, v) \in E} A[i-1, w] + l_{w, v} \end{cases}$$