Overview

**logistics:**
- pset1 out, due W10 (tomorrow) — can submit in *groups* of $\leq 3$

**last time:**
- dynamic programming on trees
- maximum independent set
- dominating set

**today:**
- shortest paths
  - with negative lengths
  - all-pairs
Shortest Paths, with Negative Lengths

**questions:**
- what is the length of the shortest path between $s$ and $t$?
- what is the length of the shortest path from $s$ to every other node?
- what happens if we get lost?
- how to deal with negative cycles?

**remarks:**
- computing the length of the shortest simple $s \rightarrow t$ path (with possibly negative lengths) is NP-hard — contains the Hamiltonian path problem

**total cost:**

$$9 + 10 + (-16 + 11 + 3) \cdot k + (-16) + 16 = 19 - 3k \rightarrow -\infty$$
**Definition**

$G = (V, E)$ directed (simple) graph, with edge length function $\ell : E \to \mathbb{Z}$.

- A **path in** $G$ is a sequence of distinct vertices $v_0, v_1, \ldots, v_k \in V$ such that $(v_i, v_{i+1}) \in E$ for all $i$. An $(s, t)$-path is a path where $v_0 = s$ and $v_k = t$.

- A **walk in** $G$ is a sequence of vertices $v_0, v_1, \ldots, v_k \in V$ such that $(v_i, v_{i+1}) \in E$ for all $i$. An $(s, t)$-walk is a walk where $v_0 = s$ and $v_k = t$.

- The **length of a walk** is the sum of the edge lengths $\sum_i \ell(v_i, v_{i+1})$.

- The **distance from $s$ to $t$ in** $G$, denoted $\text{dist}(s, t)$, is the length of the shortest $(s, t)$-walk, $\text{dist}(s, t) := \min_{(s,t)-\text{walk } w} \ell(w)$.

**remarks:**

- if $(s, t)$-walk containing a negative length cycle $\implies \text{dist}(s, t) = -\infty$

- if *no* $(s, t)$-walk containing a negative length cycle $\implies$ shortest walk is a path $\implies$ shortest walk $\leq n - 1$ edges and is of finite length
Shortest Paths, with Negative Lengths (III)

Definition

\( G = (V, E) \) directed (simple) graph, with edge length function \( \ell : E \to \mathbb{Z} \). The (single-source) shortest path problem (with negative weights) is to:

- given \( s, t \in V \), find a minimum length \((s, t)\)-path or find an \((s, t)\)-walk with a negative cycle \( \implies \text{dist}(s, t) = -\infty \)
- given \( s \in V \), compute \( \text{dist}(s, t) \) for all \( t \in V \)
- determine if \( G \) has any negative cycle

Remarks:

- Negative lengths can be natural in modelling real life
  - E.g., demand/supply on an electrical grid
  - Negative cycles manifest as arbitrage
- Negative lengths can arise as by-products of other algorithms, e.g., flows in graphs
Dijkstra’s Algorithm

Dijkstra’s algorithm: greedily grow shortest paths from source $s$
Dijkstra’s Algorithm, with Negative Lengths?

**Dijkstra’s algorithm:** greedily grow shortest paths from source \( s \)

![Graph with vertices and edges labeled with distances](image)

**remarks:**
- greedy exploration, ordering vertices \( v \in V \) by \( \text{dist}(s, v) \) — without updates!
- \( \Rightarrow \) algorithm assumes the distance only grows as the graph is explored
- \( \equiv \) assumes all edge lengths are non-negative
Lemma

$G = (V, E)$ directed (simple) graph, with edge length function $\ell : E \to \mathbb{Z}$. If $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k = t$ is a shortest $(s, t)$-walk, then

1. $s \rightarrow v_1 \rightarrow \cdots \rightarrow v_i$ is a shortest $(s, v_i)$-walk, for $i \leq k$
2. if $\ell$ is non-negative, $\text{dist}(s, v_i) \leq \text{dist}(s, v_{i+1})$ for all $i$

Proof.

(1) Cut and paste. (2) Clear.

Remarks:

- shortest walks are shortest paths, if no negative cycle
- Dijkstra’s algorithm defines subproblems by restricting the graph by $\text{dist}(s, \cdot)$
- idea: parameterize subproblems by number of edges in a walk, and allow updates to $\text{dist}(s, \cdot)$
Shortest Paths, with Negative Lengths (V)

Definition

\( G = (V, E) \) directed (simple) graph, with edge length function \( \ell : E \to \mathbb{Z} \). For \( s, t \in V \), define \( \text{dist}_k(s, t) \) to be the length of the shortest \((s, t)\)-walk using \( \leq k \) edges.

\[
\text{dist}_k(s, t) := \min_{(s, t)\text{-walk } w \mid |w| \leq k} \ell(w).
\]

Remarks:

- \( \text{dist}_k(s, t) = \infty \) if no \((\leq k)\)-edge \((s, t)\)-walk
- \( \text{dist}_0(s, s) = 0 \), \( \text{dist}_0(s, v) = \infty \) for \( v \neq s \)
Lemma

\[ G = (V, E), \ell : E \to \mathbb{Z}. \text{ Then for all } s, t \in V, \]
\[ \text{dist}_k(s, t) = \min \left\{ \text{dist}_{k-1}(s, t), \min_{v \in V} \{\text{dist}_{k-1}(s, v) + \ell(v, t)\} \right\}. \]

Proof.

Let \( s = v_0 \to v_1 \to v_2 \to \cdots \to v_j = t \) be a shortest length \( j \leq k \) \((s, t)\)-walk. Then,

- \( j < k \): hence this is a \((\leq k - 1)\)-edge \((s, t)\)-walk of length \( \text{dist}_{k-1}(s, t) \)
- \( j = k \): hence \( s = v_0 \to v_1 \to v_2 \to \cdots \to v_{k-1} \) is a shortest length \((\leq k - 1)\)-edge \((s, v_{k-1})\) walk \(\implies\) can add \( \ell(v_{k-1}, t) \) to reach \( t \)

**remark:** \( \ell(v, t) = \infty \) if there is no edge
Theorem

$G = (V, E), \ell : E \rightarrow \mathbb{Z}, s \in V$, with every vertex reachable from $s$.

1. **If there are no negative length cycles, then for all $v \in V$,
   \[ \text{dist}_{n-1}(s, v) \leq \text{dist}_n(s, v), \text{ and even } \text{dist}_{n-1}(s, v) = \text{dist}(s, v). \]

2. **If for all $v \in V$, \text{dist}_{n-1}(s, v) \leq \text{dist}_n(s, v), then there are no negative length cycles.**
Lemma

\[ G = (V, E), \ell : E \rightarrow \mathbb{Z}. \text{ Then for all } s, t \in V, \]

\[ \text{dist}_k(s, t) = \min \left\{ \begin{array}{l}
\text{dist}_{k-1}(s, t) \\
\min_{v \in V} \{\text{dist}_{k-1}(s, v) + \ell(v, t)\}
\end{array} \right\}. \]

Corollary

For all \( k \geq 0, \)

- \( \text{dist}_k(s, t) \leq \text{dist}_{k-1}(s, t) \)
- If for all \( v \in V, \text{dist}_k(s, t) = \text{dist}_{k-1}(s, t) \)

\[ \implies \text{ for all } v \in V, \text{dist}_{k+1}(s, t) = \text{dist}_k(s, t) \]

\[ \implies \text{ for all } v \in V, \text{dist}_{k+2}(s, t) = \text{dist}_{k+1}(s, t) \implies \ldots \]
Proposition

\[ G = (V, E), \ell : E \to \mathbb{Z}, s \in V, \text{ with every vertex reachable from } s. \text{ If there are no negative length cycles, then for all } v \in V, \text{ dist}_{n-1}(s, v) \leq \text{ dist}_n(s, v). \]

Proof.

Let \( s = v_0 \to v_1 \to \cdots \to v_{k-1} \to v_k = v \) be a walk of \((\leq n)\)-edges, with length \( \text{dist}_n(s, v) \).

- If \( k < n \), then this is a \((< n)\)-edge walk and hence of length \( \geq \text{dist}_{n-1}(s, v) \).
- If \( k = n \), then the walk visits \( n + 1 \) vertices \( \implies \) some vertex is repeated \( \equiv \) there is a cycle. As the cycle is of non-negative length \( C \geq 0 \), we can remove it to obtain a \((< n)\)-edge \((s, v)\)-walk of value \( d = \text{dist}_n(s, v) - C \) with \( \text{dist}_n(s, v) \geq d \geq \text{dist}_{n-1}(s, v) \).
Shortest Paths, with Negative Lengths (X)

Proposition

\[ G = (V, E), \ell : E \to \mathbb{Z}, \ s \in V, \text{ with every vertex reachable from } s. \text{ If for all } v \in V, \]
\[ \text{dist}_{n-1}(s, v) \leq \text{dist}_n(s, v), \text{ then } \lim_{k \to \infty} \text{dist}_k(s, v) \text{ is finite for all } v \in V. \]

Proof.

By previous corollary, for all \( v \in V \), \( \text{dist}_{n-1}(s, v) \geq \text{dist}_n(s, v) \implies \) for all \( v \in V \),
\[ \text{dist}_{n-1}(s, v) = \text{dist}_n(s, v) = \text{dist}_{n+1}(s, v) = \text{dist}_{n+2}(s, v) = \cdots. \]
As all \( v \) are reachable from \( s \) \( \implies \) \( \text{dist}_{n-1}(s, v) \leq \infty \) for all \( k \) and \( v \). Hence
\[ \lim_{k \to \infty} \text{dist}_k(s, v) = \text{dist}_{n-1}(s, v) \text{ is finite for all } v. \]
Proposition

\[ G = (V, E), \ l : E \to \mathbb{Z}, \ s \in V, \ \text{with every vertex reachable from } s. \ \text{If there is a} \ (s, v)\text{-walk containing a negative length cycle, then } \lim_{k \to \infty} \text{dist}_k(s, v) = -\infty. \]

Proof.

Let \( s \leadsto u \leadsto u \leadsto v \) be an \((s, v)\)-walk with length \( L \), where \( u \leadsto u \) is a negative length cycle of length \(-C < 0\). Then consider the \((s, v)\)-walk \( s \leadsto u \leadsto u \leadsto u \leadsto v \), which is of value \( L - C \). Hence, for any \( j \) there is \((s, v)\)-walk of length \( L - C \cdot j \). Hence \( \lim_{k \to \infty} \text{dist}_k(s, v) = -\infty. \)
### Proposition

$G = (V, E), \ell : E \to \mathbb{Z}, s \in V,$ with every vertex reachable from $s$. If for all $v \in V$, $\text{dist}_{n-1}(s, v) \leq \text{dist}_n(s, v)$, \( \lim_{k \to \infty} \text{dist}_k(s, v) \) is finite for all $v \in V$.

### Proposition

$G = (V, E), \ell : E \to \mathbb{Z}, s \in V,$ with every vertex reachable from $s$. If there is a $(s, v)$-walk containing a negative length cycle, then $\lim_{k \to \infty} \text{dist}_k(s, v) = -\infty$.

### Corollary

$G = (V, E), \ell : E \to \mathbb{Z}, s \in V,$ with every vertex reachable from $s$. If for all $v \in V$, $\text{dist}_{n-1}(s, v) \leq \text{dist}_n(s, v)$, then there are no negative length cycles.
Theorem

\( G = (V, E), \ell : E \to \mathbb{Z}, s \in V, \) with every vertex reachable from \( s. \)

1. If there are no negative length cycles, then for all \( v \in V, \) 
   \[ \text{dist}_{n-1}(s, v) \leq \text{dist}_n(s, v), \] 
   and 
   \[ \text{dist}_{n-1}(s, v) = \lim_{k \to \infty} \text{dist}_k(s, v) = \text{dist}(s, v). \]

2. If for all \( v \in V, \) \( \text{dist}_{n-1}(s, v) \leq \text{dist}_n(s, v), \) then there are no negative length cycles.
**Bellman-Ford**

**(single source) shortest paths:** source $s \in V$, can reach every other node

- for each $v \in V$
  - $d_0[s][v] = \infty$
- $d_0[s][s] = 0$
- for $1 \leq k \leq n$, $v \in V$
  - $d_k[s][v] = d_{k-1}[s][v]$
  - for $u \in N^-(v)$
    - $d_k[s][v] = \min\{d_k[s][v], d_{k-1}[s][u] + \ell(u,v)\}$
- for $v \in V$
  - if $d_n[s][v] < d_{n-1}[s][v]$
    - return ‘‘negative cycle detected’’
  - return $d_{n-1}[s][\cdot]$

**correctness:** clear

**complexity:**

- **time**
  - clearly $O(n^3)$
  - better: $O(mn)$, $d_k[s][\cdot]$ updates along edges

- **space**
  - clearly $O(n^2)$
  - better: only store $d_{cur}[s][\cdot]$ and $d_{prev}[s][\cdot] \implies O(n)$


Bellman-Ford (II)

**remarks:**

- compute actual paths by storing pointers indicating how \(d_k[s'][\cdot]\) was updated, e.g.,
  \[v_{k-1} = \arg\min_{u \in V} \{\text{dist}_{k-1}(s, u) + \ell(u, v_k)\}\ .\]

- detecting negative cycles
  - Bellman-Ford will detect any negative cycles reachable from \(s\) in \(G\)
    \[\implies\] one Bellman-Ford call per vertex will detect if there is any negative cycle in \(G\)
    \[\implies\] \(O(mn^2)\) time
  - *better:* consider \(G' = (V \cup \{s'\}, E \cup \{(s', v)\}_{v \in V})\) with \(\ell'(s', v) = 0\)
    \[\implies\] all negative cycles in \(G\) are reachable from \(s'\) in \(G'\)
    \[\implies\] one Bellman-Ford required \(\implies\) \(O(mn)\) time

- directed acyclic graphs
  - no (negative) cycles
  - can simplify Bellman-Ford so \(\text{dist}_k(s, \cdot)\) only updates \(v_k\), according to topological ordering \(v_1 \prec v_2 \prec \cdots \prec v_n\) — yields Dijkstra-esque algorithm
    \[\implies\] \(O(m + n)\) time (**exercise**)
All-Pairs Shortest Paths

Definition

\( G = (V, E) \) directed (simple) graph, \( \ell : E \rightarrow \mathbb{Z} \). The **shortest path problem** is to:

- given \( s, t \in V \), find a minimum length \((s, t)\)-path
- given \( s \in V \), compute \( \text{dist}(s, t) \) for all \( t \in V \) (single-source)
- compute \( \text{dist}(s, t) \) for all \( s, t \in V \) (all pairs)

**single-source:**

- **Dijkstra:**
  - non-negative lengths
  - \( O((m + n) \log n) \) time (heaps), \( O(m + n \log n) \) (Fibonacci heaps)

- **Bellman-Ford:**
  - arbitrary weights
  - \( O(mn) \) time
### All-Pairs Shortest Paths (II)

#### Definition

A directed (simple) graph $G = (V, E)$ and a length function $\ell : E \to \mathbb{Z}$. The **shortest path problem** is to:

- given $s, t \in V$, find a minimum length $(s, t)$-path
- given $s \in V$, compute $\text{dist}(s, t)$ for all $t \in V$ (single-source)
- compute $\text{dist}(s, t)$ for all $s, t \in V$ (all pairs)

**all-pairs:**

- $n$ runs of *Dijkstra*:
  - non-negative lengths
  - $O(n \cdot (m + n) \log n)$ time (heaps), $O(n \cdot (m + n \log n))$ (Fibonacci heaps)

- $n$ runs of *Bellman-Ford*:
  - arbitrary weights
  - $O(n \cdot mn)$ time $\implies \Theta(n^4)$ if $m = \Theta(n^2)$

**question:** can we do better?
idea: use a new parameterization of the subproblems

Definition

\( G = (V, E) \) directed (simple) graph, with edge length function \( \ell : E \rightarrow \mathbb{Z} \). Order \( V \) as \( v_1 \prec v_2 \prec \cdots \prec v_n \). A \((u, v)\)-walk \( u = w_0 \rightarrow w_1 \rightarrow \cdots \rightarrow w_i = v \) has **intermediate index** \( \leq j \), if \( w_1, \ldots, w_{i-1} \in \{v_1, \ldots, v_j\} \). For \( s, t \in V \), define \( \text{dist}^k(s, t) \) to be the length of the shortest \((s, t)\)-walk of intermediate index \( \leq k \).

\[ \begin{align*} 
\text{dist}^0(v_3, v_4) &= \ell(v_3, v_4) = 8 \\
\text{dist}^1(v_3, v_4) &= 5 \\
\text{dist}^2(v_3, v_4) &= 4 
\end{align*} \]
Lemma

\( G = (V, E), \ell : E \to \mathbb{Z}, \text{ with no negative cycles. Then for all } s, t \in V, \)
\( \text{dist}^0(s, t) = \ell(s, t), \text{ and} \)
\[ \text{dist}^k(s, t) = \min \left\{ \text{dist}^{k-1}(s, t), \text{dist}^{k-1}(s, v_k) + \text{dist}^{k-1}(v_k, t) \right\}. \]

Proof.

Let \( s = w_0 \to w_1 \to w_2 \to \cdots \to w_i = t \) be a shortest length \((s, t)\)-walk of intermediate index \( \leq k \) and length \( \text{dist}^k(s, t) \). There are two cases:

- **index < k:** hence is of value \( \text{dist}^{k-1}(s, t) \)
- **index = k:**
  - no negative cycles \( \implies \) shortest walk is *path* \( \implies \) \( v_k \) appears exactly once
  \( \implies \) \( s \rightsquigarrow v_k \) path and \( v_k \rightsquigarrow t \) path are of index < k, and must be shortest paths
Floyd-Warshall

for $1 \leq i, j \leq n$
$d^0[i][j] = \ell(i, j)$

for $1 \leq k \leq n$
  for $1 \leq i, j \leq n$
    
    \[
    d^k[i][j] = \min \left\{ d^{k-1}[i][j], d^{k-1}[i][k] + d^{k-1}[k][j] \right\}
    \]

for $1 \leq i \leq n$
  if $d^n[i][i] < 0$
    return ‘‘negative cycle detected’’

**complexity:**

- $O(n^3)$ time
- space
  - clearly $O(n^3)$
  - better: only store $d^{\text{cur}}[\cdot][\cdot]$ and $d^{\text{prev}}[\cdot][\cdot] \rightarrow O(n^2)$

**correctness:**

- if no negative cycles, correctness is clear
- if some negative cycle, ???

**remarks:**

- compute actual paths by storing pointers indicating how $d^k[\cdot][\cdot]$ was updated
Floyd-Warshall (II)

**Proposition**

\[ G = (V, E), \ell : E \to \mathbb{Z}, \text{with some negative cycle. Then the Floyd-Warshall}\]

algorithm correctly detects this cycle.

**Proof.**

Let \( k \leq n \) be the minimum index of a negative length cycle

\[ k = \min_{\text{negative length } C} \max_{i : v_i \in C} i. \]

Pick such a cycle \( C \), where \( C \) is

\[ v_k = w_0 \to w_1 = v_i \to \cdots \to w_j = v_k. \]

By choice of \( k \),

\[
\begin{align*}
\text{d}^{k-1}[k][i] &= \text{dist}^{k-1}(k, i) \leq \ell(w_0, w_1) \\
\text{d}^{k-1}[i][k] &= \text{dist}^{k-1}(i, k) \leq \ell(w_1, w_2) + \cdots + \ell(w_{j-1}, w_j) \\
\Rightarrow \quad \text{d}^k[k][k] &\leq \text{d}^{k-1}[k][i] + \text{d}^{k-1}[i][k] = \ell(w_0, w_1) + \cdots + \ell(w_{j-1}, w_j) = \ell(C) < 0 \\
\Rightarrow \quad \text{d}^{k+1}[k][k] &\leq \text{d}^k[k][k] < 0 \\
\Rightarrow \quad \text{d}^n[k][k] < 0 &\Rightarrow \text{negative cycle detected}
\end{align*}
\]
Floyd-Warshall

for \(1 \leq i, j \leq n\)

\[d^0[i][j] = \ell(i, j)\]

for \(1 \leq k \leq n\)

for \(1 \leq i, j \leq n\)

\[d^k[i][j] = \min\left\{d^{k-1}[i][j], d^{k-1}[i][k] + d^{k-1}[k][j]\right\}\]

for \(1 \leq i \leq n\)

if \(d^n[i][i] < 0\)

return ‘‘negative cycle detected’’

remarks:

- compute actual paths by storing pointers indicating how \(d^k[\cdot][\cdot]\) was updated

complexity:

- \(O(n^3)\) time
- space
  - clearly \(O(n^3)\)
  - better: only store \(d^{\text{cur}}[\cdot][\cdot]\) and \(d^{\text{prev}}[\cdot][\cdot] \implies O(n^2)\)

correctness:

- if no negative cycles, correctness is clear
- if some negative cycle, correctness is now done
**Overview (II)**

**logistics:**
- pset1 out, due W10 (tomorrow) — can submit in *groups* of $\leq 3$

**today:**
- shortest paths
  - with negative lengths — Bellman-Ford in $O(mn)$ time
  - all-pairs — Floyd-Warshall in $O(n^3)$ time

**next time:**
- *more* dynamic programming