Dijkstra's algorithm:

Shortest path from s to all other vertices

But assumed positive edge lengths.

What if there are negative weight edges? (or negative lengths)

Is the problem still well-defined?

Example:

What's the shortest path from a to f?

The cycle b → c → d → e → b

has total weight -1

So it's called a negative weight cycle.

Our goal: Given a graph with edge weights, determine if there is a negative weight cycle. If there are none, then determine for all pairs of vertices the shortest path length between them.
For $G = (V,E)$, let $V = \{1, 2, ..., n\}$ so the vertices are numbered $1, 2, ..., n$.

**Goal:** find $n \times n$ matrix $D$ where $D(i,j)$ = length of the shortest path from $i$ to $j$.

**Dijkstra’s algorithm: high-level idea:**

$\text{dist}(s) = 0$

for all $w \neq S$, $\text{dist}(w) = \infty$

$R = \emptyset$

while $R \neq V$

let $w$ be vertex not in $R$ with min $\text{dist}(w)$

Add $w$ to $R$

Update $\text{dist}(z)$ for all $(w,z) \in E$

Thus: $\text{dist}(z)$ = length of shortest path from $s$ to $z$ only using vertices in $R$ as intermediate vertices.
Back to the general problem, let's assume no negative weight cycles for now, & let's compute D.

We'll use dynamic programming.

Dijkstra's:
\[ \text{dist}(z) = \text{length of shortest } s \rightarrow z \text{ path only using } R \text{ as intermediate vertices} \]

Here: use prefix of input so 31, ..., k_j instead of R

For \( 0 \leq k \leq n \), \( 1 \leq i \leq n \), \( 1 \leq j \leq n \),
let
\[ f_k(i, j) = \text{length of shortest path from } i \text{ to } j \]
only using vertices in \( 31, ..., k_j \) as intermediate vertices.

Our goal: compute \( C_n = D \).
Base case: $k=0$

$$C_0(i,j) = \begin{cases} \omega(i,j) = \text{weight of edge } i \rightarrow j & \text{if } ij \in E \\ \infty & \text{if } ij \notin E \end{cases}$$

For $k > 0$

either the shortest $i \rightarrow j$ path using

$1)$ uses $k$ as an intermediate vertex

or $2)$ doesn't use $k$.

Case 2:

$$C_k(i,j) = C_{k-1}(i,j)$$

Case 1:

$$C_k(i,j) = C_{k-1}(i,k) + C_{k-1}(k,j)$$
Therefore, for $k \geq 1$:

$$C_k(i,j) = \min \{ C_{k-1}(i,j), C_{k-1}(i,k) + C_{k-1}(k,j) \}$$

**Floyd-Warshall $(G,w)$**

**Input:** directed $G = (V,E)$ with $V = \{1, \ldots, n\}$

& for $ij \in E$ \( \omega(i,j) \) = weight of edge $i \rightarrow j$

**Output:** matrix $D$

for $i = 1 \rightarrow n$

for $j = 1 \rightarrow n$

if $ij \in E$ then \( C(i,j,0) = \omega(i,j) \)

else \( C(i,j,0) = \infty \)

if $i = j$ then \( C(i,i,0) = 0 \)

for $k = 1 \rightarrow n$

for $i = 1 \rightarrow n$

for $j = 1 \rightarrow n$

\( C(i,j,k) = \min \{ C(i,j,k-1), C(i,k,k-1) + C(k,j,k-1) \} \)

Return ($C(\cdot,\cdot, n)$)
Running time: $O(n^3)$

What about negative weight cycles?

Suppose vertex $i$ is on a negative weight cycle $C$.

Then there is a path from $i$ to $i$ of negative weight.

Hence, $C(i,i,n) < 0$

So look at diagonal entries,

if there is an $i$ where $C(i,i,n) < 0$

then there is a negative weight cycle which includes vertex $i$

else if all diagonal entries = 0

then no negative weight cycles

& $C(i,j,n) = \text{length of shortest } i \rightarrow j$ path

$= D(i,j)$. 