Floyd–Warshall algorithm

Recall: Dijkstra's Alg.:

- finds shortest paths from one start vertex to all others.
  (Single-Source Shortest Paths problem, SSSP)
- only works if all edges have nonnegative weights.

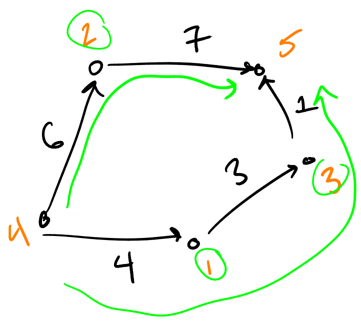
F-W alg:

- Works w/ negative weights
- Solves the All-Pairs Shortest Paths (APSP)

↳ ie. want an array d where $d[u,v]$ = length of shortest path $u \to v$.

Add a parameter:

$d[u,v,k]$ = length of shortest path $u \to v$ which only uses vertices $1..k$ as intermediate steps.



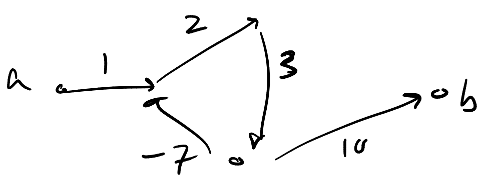
$d[4,5,2] = 13$

$d[4,5,3] = 8$

$d[4,5,1] = \infty$

$d[4,4,1] = 0$

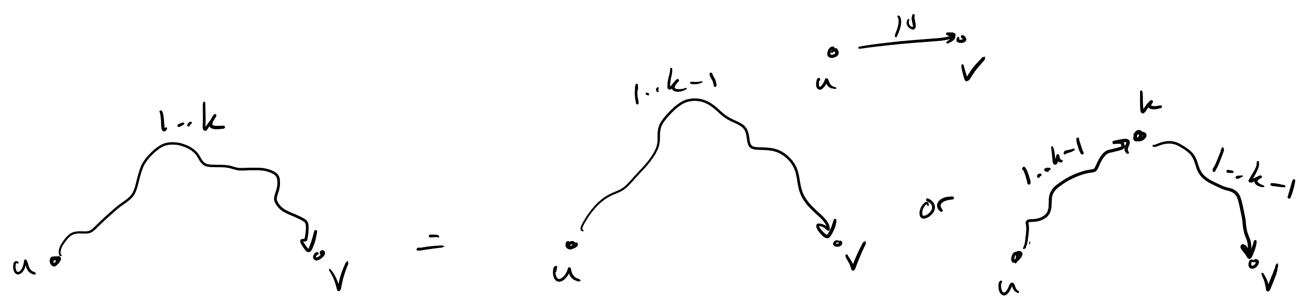
Base cases



negative cycles !!

$$d[u,u,0] = 0$$

$$d[u,v,0] = \begin{cases} \infty & \text{if no edge } u \to v \\ w_{uv} & \text{if edge } u \to v \end{cases}$$



$$d[u,v,k] = \min\left(d[u,v,k-1],\ d[u,k,k-1] + d[k,v,k-1] \right)$$

Careful in practice — ∞ might overflow.

Actually can get away with just storing everything in a single $n \times n$ array.

initialize $d[u,v]$ using base cases

```
for k in 1..n:
    for u in 1..n:
        for v in 1..n:
            d[u,v] = min(d[u,v], d[u,k] + d[k,v])
```

$\Theta(n^3)$! ie V^3.

Dijkstra: $\Theta(E \lg V)$. Which could be $O(V^2 \lg V)$.

Any vertex u is contained in a negative cycle iff $d[u,u] < 0$.

How to find actual shortest paths?

- Could remember optimal choices + reconstruct
- Better way:

  let $next[u,v]$ = next vertex along shortest path $u \to v$.

  initialize $next[u,u] = u$.

  $next[u,v] = v$ if edge $u \to v$

  undefined otherwise.

  in induction step, if $d[u,v,k] = d[u,v,k-1]$, $next[u,v]$ does not change

  if $d[u,v,k] = d[u,k,k-1] + d[k,v,k-1]$,

  then set $next[u,v] = next[u,k]$.