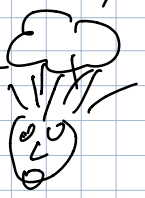
HW8 out today

Midterm 2 — Conflict



3 wks

Review

Final May 9

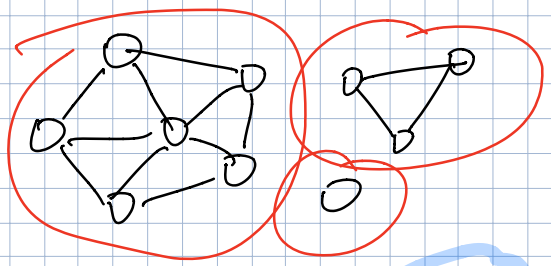
Whatever-first search — traversal / reachability

Depth-first — detecting cycles
- topological sort
- dynamic programming
- strong components

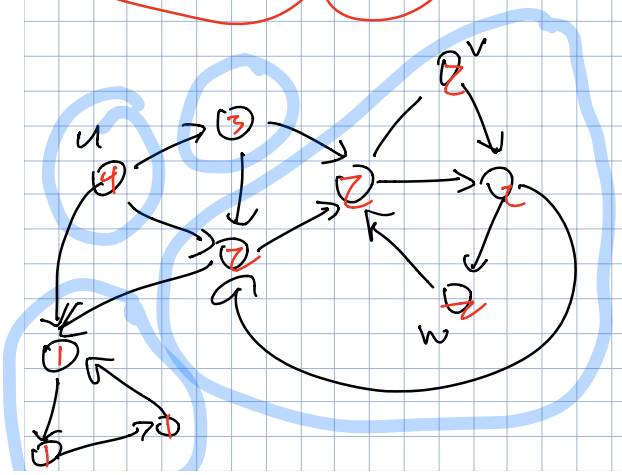
Breadth-first — shortest paths

weights → Dijkstra's ← greedy

Bellman-Ford ← dyn. prog.



3 components



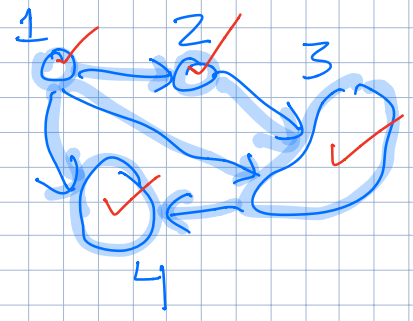
u can reach v

but v can't reach u

v + w strongly connected

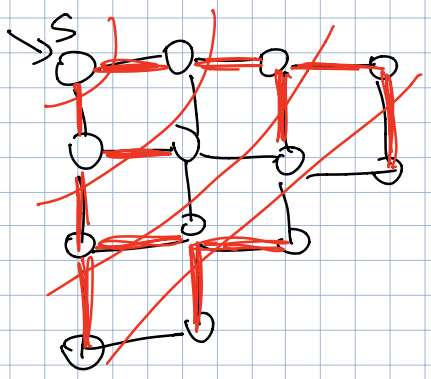
4 strong components

condensation scc-graph scc(G)



dag!

Kosaraju-Sharir
Tarjan $O(V+E)$



BFS — expanding wavefront

shortest paths

$O(V+E)$ time

length = #edges

Input = directed graph $G=(V,E)$

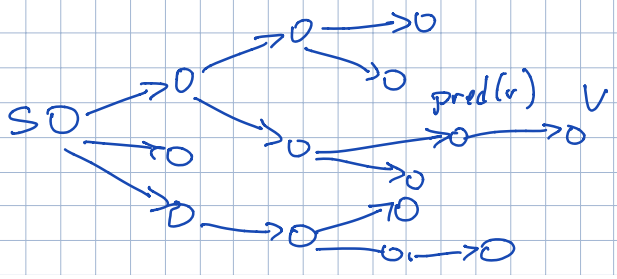
lengths $\ell: E \to \mathbb{R}$

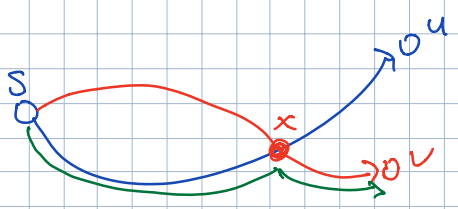
vertices s + t

Output: ~~path~~ walk $s = v_0 \to v_1 \to v_2 \to \cdots \to v_k = t$

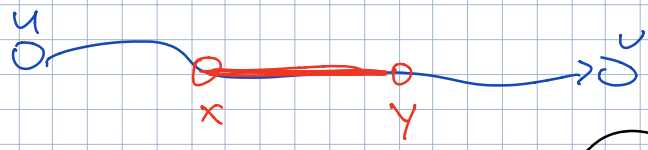
minimize $\sum_{i=1}^{k} \ell(v_{i-1} \to v_i)$

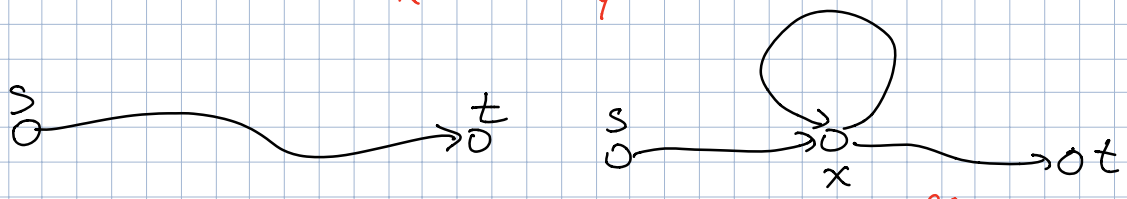
SSSP – Find shortest paths from s to every other vertex.





Subpath of shortest path is a shortest path





Impossible if $\ell(e) \geq 0$ or no neg. cycles

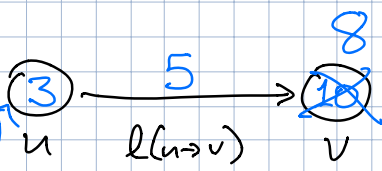
Neg cycle ⇒ shortest walk doesn't exist

NO NEGATIVE CYCLES

We want pred for every vertex (except s)
dist for every vertex

Init:

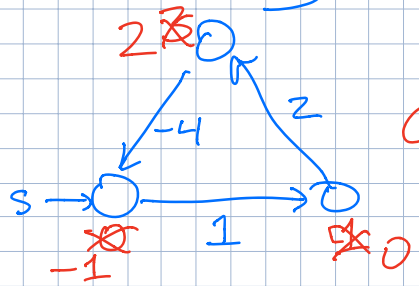
```
dist(s) ← 0
for all v≠s
    dist(v) ← ∞
for all v
    pred(v) ← NULL
```



$u \to v$ is tense

```
if dist(u) + ℓ(u→v) < dist(v)
    dist(v) ← dist(u) + ℓ(u→v)
    pred(v) ← u
```

relaxing



∞ loop!

What order? Which tense edges to relax?

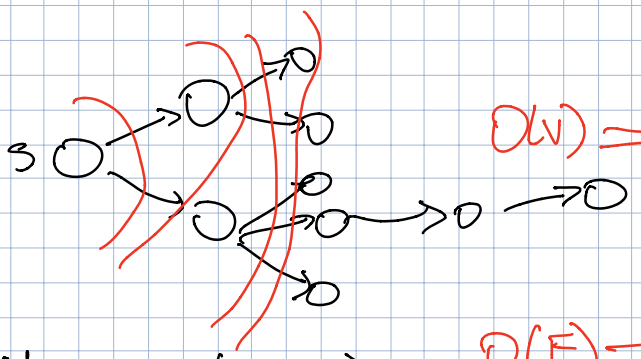
stack – $\Theta(2^V)$

queue Bellman Ford $O(VE)$

heap Dijkstra $\begin{cases} O(E \log V) & \text{no neg } \ell \\ \Theta(2^V) & \text{neg } \ell \end{cases}$

DAG — $O(E+V)$!

Dijkstra: $\boxed{\ell > 0}$



No neg edges ⇒

each vertex is EXTRACTED ≤ once

We extract verts in increasing dist(v)

put s in PQ with key 0

while heap not empty

$O(V)$ ⇒ v ← EXTRACT MIN

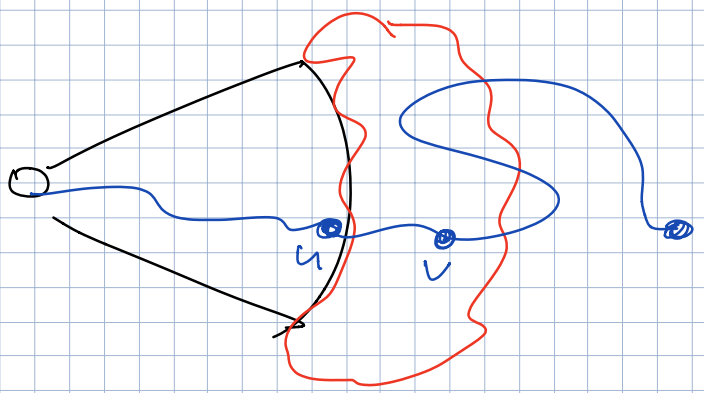
for all v→w

$O(E)$ ⇒ if v→w is tense

relax v→w

Insert(w, dist(w))

or UPDATE



Binary heap:

$O((E) \log V)$

Fibonacci heap

$O(E + V \log V)$