CS3000
6/10 - Tues.

Admin

- HW5 due 6/12 9pm
- APP8 due 11:30 6/11
- exam #2 $\boxed{6/12}$
- gradchance due 6/18 9pm
- XC exam question 6/18 during lecture

Agenda

1. APSP
2. Optimization Approach
3. Floyd-Warshall

1. All Pairs Shortest Paths

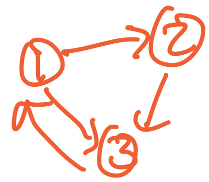
shortest paths, so far...

- BFS
- Dijkstra's

} SSSP greedy
for every vertex v,
v.π (pred)

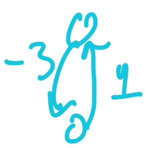
→

APSP



shortest path
1 ⇝ 2
1 ⇝ 3
2 ⇝ 3
2 ⇝ 1
3 ⇝ 2
3 ⇝ 1

Assumptions:
- graph is directed, weighted
- neg weight edges ok!
- cycles are ok!
- no neg weight cycles ☆



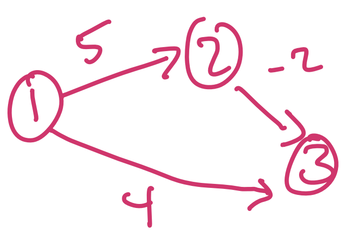
Before... for each vertex v
- v.d distance from source
- v.π predecessor on path

→ value of optimal soln
→ construct actual optimal soln

Focus: distance values
- many v.d's for a given vertex
  v.d for vertex 1, 2, 3, ..., |V|

graph G is adjacency matrix $G.A[u][v]$ → weight of edge u, v
- vertices are numbered 1, 2, ..., |V|

(ex) • focus on final answer, not algorithm!



Distances

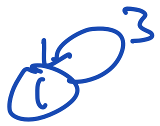
	1	2	3
1	0	5	3
2	∞	0	-2
3	∞	∞	0

Predecessors

	1	2	3
1	nil	1	2
2	nil	nil	2
3	nil	nil	nil

- vertex goes to itself
  1,1 2,2 3,3 } dist 0 π nil
- no path exists   } dist ∞ π nil

ex
neg weight cycle



ok!

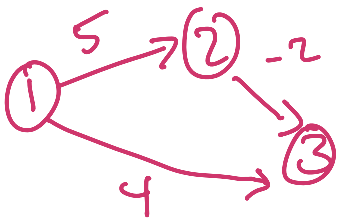
2. optimization Approach

↳ final answer: D table
$|V| \times |V|$

d_{ij} = distance of s.p. from vertex i to vertex j

optimization: greedy not great :(
DP!

BF → D+C → DP → greedy → graphs!
BF → data str



How did we build table above?

1,2   edge (1,2)   dist = 5
2,3   edge (2,3)   dist = -2
1,3   edge (1,3) ?   dist = 4
try 1,2,3   dist = 3
↳ contain s.p. 1 ⇝ 2
s.p. 2 ⇝ 3 } optimal substructure!

Ask ourselves...
- can we get i ⇝ j on one edge (i,j)
- can it be shorter?
  i ⇝ 1 ⇝ j
  i ⇝ 2 ⇝ j   all possible intermediate vertices
  i ⇝ 3 ⇝ j   use if better than what we had already
  ....

In practice... many D tables

$D^{(k)}$ is: kth D table
should I go through vertex k

$d_{ij}^{(k)}$ = distance from vertex i to vertex j
(best so far)
better to go $i \rightsquigarrow k \rightsquigarrow j$ or keep what we had

$D^{(0)}$ is first table (weights table)

$D^{(n)}$ is final table

$$d_{ij}^{(0)} = \begin{cases} 0 & \text{if } i=j \\ \infty & \text{if } (i,j) \notin E \\ w(i,j) & \text{if } (i,j) \in E \end{cases}$$

$d_{ij}^{(k)}$ $i \rightsquigarrow k \rightsquigarrow j$?

$k-1$ = prev table

Recursive formula

$$d_{ij}^{(k)} = \min\Big(\underbrace{d_{ij}^{(k-1)}}_{\text{keep what we had}},\ \underbrace{d_{ik}^{(k-1)} + d_{kj}^{(k-1)}}_{\text{go through } k}\Big)$$

3. Floyd Warshall

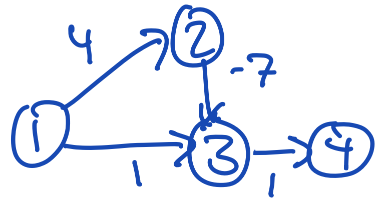
run-time $\Theta(V^3)$

FLOYD-WARSHALL(G, w)

```
1  let D^(0) be a new V × V matrix
2  for i = 1 to V
3      for j = 1 to V
4          if i == j
5              d_ij^(0) = 0
6          elseif (i, j) ∈ G.E
7              d_ij^(0) = w(i, j)
8          else
9              d_ij^(0) = ∞
10 for k = 1 to V
11     let D^(k) be a new V × V matrix
12     for i = 1 to V
13         for j = 1 to V
14             d_ij^(k) = min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
15 return D^(V)
```

Lines 1–9: $D^{(0)}$

weight of edge i,j if exists
∞ if no edge i,j
0 if $i=j$



$D^{(0)}$

	1	2	3	4
1	0	4	1	∞
2	∞	0	-7	∞
3	∞	∞	0	1
4	∞	∞	∞	0

lines 10 → 14

- $k = 1, 2, 3, 4$
- for each k, make $D^{(k)}$
- better to go $i \rightsquigarrow k \rightsquigarrow j$?

↳ keep $d_{ij}^{(k-1)}$ or $d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$

$D^{(1)}$ for every i,j better to keep $D_{ij}^{(0)}$ or go through vertex 1?

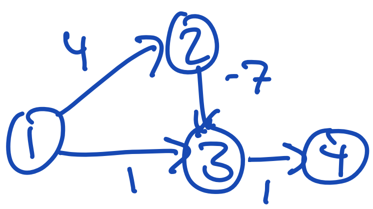
	1	2	3	4
1	0	4	1	∞
2	∞	0	-7	∞
3	∞	∞	0	1
4	∞	∞	∞	0

- $d_{ij}^{(0)}$ is always the answer for $d_{ij}^{(1)}$

$D^{(2)}$ for every i,j better to keep $D_{ij}^{(1)}$ or go through 2?

	1	2	3	4
1	0	4	-3	∞
2	∞	0	-7	∞
3	∞	∞	0	1
4	∞	∞	∞	0

- $d_{13}^{(1)}$ replaced with $d_{12}^{(1)} + d_{23}^{(1)}$



$D^{(3)}$ for every i,j better to keep $D_{ij}^{(2)}$ or go through vertex 3?

	1	2	3	4
1	0	4	-3	-2
2	∞	0	-7	-6
3	∞	∞	0	1
4	∞	∞	∞	0

- $d_{14}^{(2)}$ is replaced with $d_{13}^{(2)} + d_{34}^{(2)}$
- $d_{24}^{(2)}$ is replaced with $d_{23}^{(2)} + d_{34}^{(2)}$

$D^{(4)}$ same as $D^{(3)}$

done!