CS3000
6/8 – Thurs :)

Admin

- Short HW3 due today 9pm
- Long HW5 out now, due tues 9pm (last long HW :))
- Fun recitation today

Agenda

1. All pairs shortest paths
2. Floyd Warshall algorithm
3. Exam #2 logistics, second chance HWs

Recap

Dijkstra's ~ restriction on graph type

- directed
- weighted
- non-neg edge weights
- cycles ok

Relax an edge (u,v)

- Can we improve distance to v by going through u?

Vertex attributes?

- $v.d$ ~ shortest distance to v from s
- $v.\pi$ ~ predecessor vertex on shortest path

11. APSP

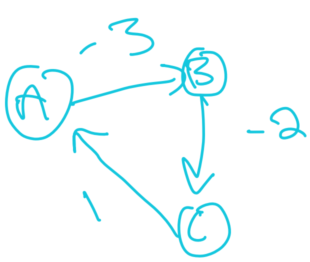
SSSP → APSP
(Dijkstra's) (Floyd Warshall)

- for every pair of vertices u and v, find shortest path from u to v

- directed
- weighted
- negative weights ok!
- cycles ok!

⤳ $\boxed{\text{negative weight cycles not ok}}$

(ex) negative weight cycles bad in $\boxed{SP}$



SP from A to B...

A → B -3
A → B → C → A → B -7
→ C → A → B -11

SSSP ~ BFS, Dijkstra's

v.d
v.π ⤳ shortest path from s to v

For every vertex, care about one path

now... APSP

For every vertex, we care about |V| paths

- care about v.d from A

v.d from B

v.d from C

⋮

v.d from all vertices

- care about v.π from A

  v.π from B

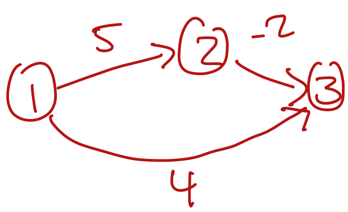
  ⋮

  v.π from all vertices

Represent d, π values in a table

- graph G stored in adj matrix

ex



D table $V \times V$

	1	2	3
1	0	5	3
2	∞	0	-2
3	∞	∞	0

Goal

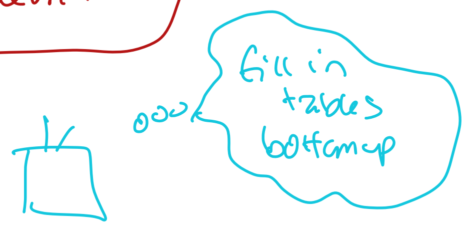
π table $V \times V$

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

APSP is an optimization problem

- use Dynamic Programming!
- Build D tables, π tables

- update based on previous values
- How can we use solutions to smaller problems to solve the bigger problem?



Build many D tables

- $D^{(k)}$ is the k^{th} D table
- has entry for every pair of vertices
- $d_{ij}^{(k)}$ → one entry in k^{th} D table
  → distance from i to j (so far...)

$\Pi^{(k)}$ is kth π table

$D^{(0)}$ table (base case)

- don't know paths yet
- only edges so far

$$d_{ij}^{(0)} = \begin{cases} 0 & \text{if } i == j \\ \infty & \text{if } i \neq j \text{ and no edge } (i,j) \\ w(i,j) & \text{otherwise} \end{cases}$$

$\Pi^{(0)}$ table (base case)

- if (i,j) exists, $\pi_{ij}^{(0)} = i$

Starting from $D^{(0)}$, want to fix:

- there might be a shorter path $i \rightsquigarrow j$ that doesn't go through (i,j)
- there might be a path $i \rightsquigarrow j$ even if (i,j) does not exist

2. Floyd Warshall

- just D tables
- when we build $D^{(k)}$, we try to improve $d_{ij}^{(k-1)}$
- does it help to go through vertex k?

$i \rightsquigarrow j$ vs. $i \rightsquigarrow k \rightsquigarrow j$

$\rightsquigarrow$ = path

Recursive Formula $\longrightarrow$ Bottom-up code

$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

Is it better to go through k?

```
FLOYD-WARSHALL(G, w)
let D^(0) be a new V × V matrix
for i = 1 to V
    for j = 1 to V
        if i == j
            d_ij^(0) = 0
        elseif (i, j) ∈ G.E
            d_ij^(0) = w(i, j)
        else
            d_ij^(0) = ∞
for k = 1 to V
    let D^(k) be a new V × V matrix
    for i = 1 to V
        for j = 1 to V
            d_ij^(k) = min(d_ij^(k−1), d_ik^(k−1) + d_kj^(k−1))
return D^(V)
```

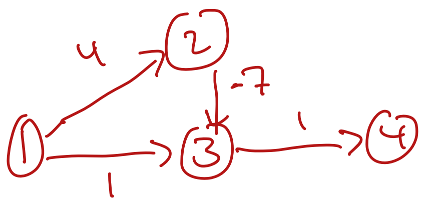
$D^{(0)}$

Recursive formula, implemented bottom-up

$\Theta(V^3)$

10:48

Floyd-Warshall example



$D^{(0)}$ (just edges)

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 4 & 1 & \infty \\ 2 & \infty & 0 & -7 & \infty \\ 3 & \infty & \infty & 0 & 1 \\ 4 & \infty & \infty & \infty & 0 \end{array}$$

$D^{(1)}$

Can we do better by going through ①?

$d_{ij}^{(0)}$ vs. $d_{i1}^{(0)} + d_{1j}^{(0)}$

$\infty \quad \infty$

$D^{(1)} = D^{(0)}$

no thing changes!

$D^{(2)}$

go through 2?

yes! $1 \to 3$

$1 \to 2 \to 3$

$d_{ij}^{(1)}$ vs. $d_{i2}^{(1)} + d_{2j}^{(1)}$

$D^{(2)}$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 4 & -3 & \infty \\ 2 & \infty & 0 & -7 & \infty \\ 3 & \infty & \infty & 0 & 1 \\ 4 & \infty & \infty & \infty & 0 \end{array}$$

$D^{(3)}$

Can we go through 3

yes!

$1 \rightsquigarrow 3 \to 4$

$2 \to 3 \to 4$

$d_{ij}^{(2)}$ vs. $d_{i3}^{(2)} + d_{3j}^{(2)}$

$D^{(3)}$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 4 & -3 & -2 \\ 2 & \infty & 0 & -7 & -6 \\ 3 & \infty & \infty & 0 & 1 \\ 4 & \infty & \infty & \infty & 0 \end{array}$$

distances

$D^{(4)}$

Can we go through 4?

no!

done!

$D^{(4)} = D^{(3)}$ = final table

3. Exam + Second Chance HW

- exam #2 $\boxed{\text{6/15 during class}}$
  - on paper
  - 8.5x11in cheat sheet, one side
  - 90-min exam
- Topics
  - greedy
  - amortized
  - DFS
  - heapsort
  - BFS
  - topo
  - MST
  - Dijkstra (SSSP)
  - APSP
  - max flow
- Practice
  - problems out Sat
  - Solutions out mon/tues.
    - ↳ go through in recitation
- Types of Questions
  - what would greedy choice be?
  - argue it's optimal
  - write pseudo code
  - which greedy choice is better?

greedy

- what does heapsort do?
- agg. analysis on sequence of n operations
- vs. worst case

Graph Questions

- what does this do on this graph?
- modify known algo to work on
  a different type of graph
  or
  produce a different output
- utility code in the exam
- write pseudocode to do something
  new
- run time of a given algorithm

Second chance HWs

- one long
- one short

resubmit for new grade
(except short 4)

sep. on gradescope

- update original submission grade

Deadline

6/20 9pm

no late submissions