COP3503 4/9/24

1. Floyd-Warshall's 4/9
2. Hashing 4/11
3. MCM + Edit Distance Steinberg 4/16
4. Final Exam Rev 4/8 Saathwika

TODAY

0. PS checker, strategy
1. On paper All-Pairs Shortest Path
   - Floyd-Warshall
2. Computer: my posted code
   examples: bigtruck (kattis)
   12 yrs a slave (FITSPS)

PS strategy

lisinopril → lis ?

- numDrugs-1
  - other drugs
- numCodes-1
  - other codes

Just make this graph

flow == numDrugs-1 ?

All Pairs Shortest Path

Directed Weight Graph

rerun Dijkstra's $|V|$ times, where V is set of vertices

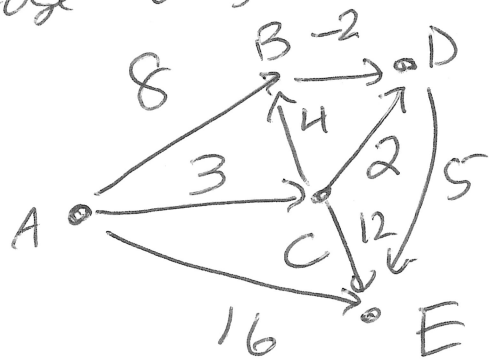
For dense graphs, this will work equally but it also handles neg. edge weights. (Dijkstra's V times $O(E \cdot V \lg V)$

FYI, Bellman-Ford does single source shortest path + allows for negative edge weights. (Can use to look for negative weight cycles) — $O(VE)$, run V times would be $O(V^2E)$

Idea behind Floyd-Warshall's:

Introduce intermediate vertices one by one.

Init shortest paths are adj[u][v] → the edge weight btw vertex u and vertex v



	A	B	C	D	E
A	0	8	3	∞	16
B	∞	0	∞	-2	∞
C	∞	4	0	2	12
D	∞	∞	∞	0	5
E	∞	∞	∞	∞	0

Allow child to go through vertex B. (intermediate k)

Old path = dist[i][j]

New path = dist[i][k] + dist[k][j]

if smaller ⟹ update

dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j]);

dist[A][D] or dist[A][B] + dist[B][D]

∞ 8 + -2

6

~~dist[A][B]~~ 8.

vertex C

8

dist[A][B] dist[A][C] + dist[C][B]

8 3 + 4

7

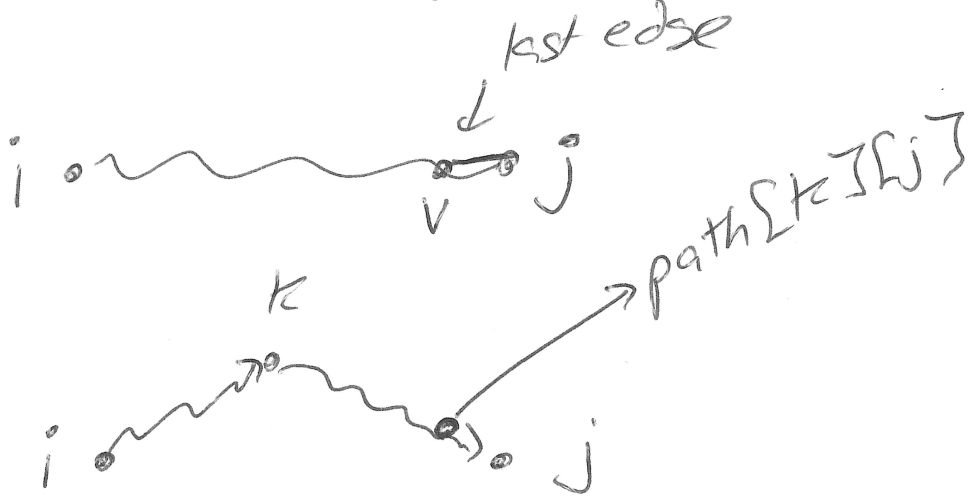
Code

```
for (int k=0; k<n; k++) // Intermediate v.
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            d[i][j] = min(d[i][j], d[i][k] + d[k][j]);
```

$O(V^3)$

Path reconstruction

Let path[i][j] = last vertex on the shortest path from i to j right before getting to vertex j.




```
if (d[i][k] + d[k][j] < d[i][j]) {
    d[i][j] = d[i][k] + d[k][j];
    path[i][j] = path[k][j];
}
```

path[2][0-4] = [2, 0, 2, 1, 2]

path from 2→3 : 2→0→1→3

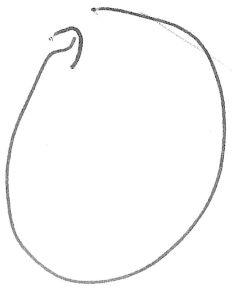
path from 2→1 :

path from 2→0 :

build back the answer from destination back to the source.

Neg cycle detection

If there is a negative cycle, that cycle should be no more then n edges



if there was a neg weight cycle for some vertex, i.

if $d[i][mid] + d[mid][i] < 0$ neg cycle

Not fast but at end ~~algorithm~~ do this

check for all pairs i, mid.

if $d[i][i] < 0 \Rightarrow$ neg weight cycle

BIGTRUCK IDEA

Distance: $10^5 \times$ edge + ~~(100 - goodies)~~ not going work

obj w/ 2 item breaks ties

$10^5 \times$ edge weight − goodies