PS 15

Linxi Li

24.4-6 simply we ~~a~~

$x_i - x_j \le b_k$ and $x_j - x_i \le -b_k$, and then solve this Problem like

~~modified Bellman-ford~~ usual Bellman-ford Problem

25.1-6

for each vertex, we would need to compute the Predecessor when $j \ne i$.

we need to compute $L_{i,k} + w(k,j) = L[i,j]$.

Since we need $O(n)$ vertexes, and for each vertexs we need $O(n-1)$

And $O(n-1)$ to compute for each node.

total time $O(n^3)$

25.2-2

$W_{ij} = 1$ if there's edge between i, j

$W_{ij} = 0$ if there's no direct edge

we simply

modified-shortest-Paths (L, W)

$n = L.rows$

let $L' = (l'_{ij})$ be new $n \times n$ matrix

for $i = 1$ to n:

for $j = 1$ to n.

$l'_{ij} = \infty$

for $k = 1$ to n.

~~$l'_{ij} = \min(l'_{ij}, l_{ik} + w$~~

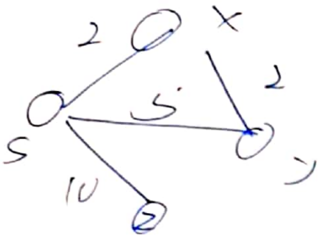
$l''_{ij} = l'_{ij} \vee (l_{ik} \wedge w_{kj})$

SLOW-ALL-PAIRS-shortest-PATHS (L, W).

25-3-4.

I think the Problem is that it changes the shortest Paths

for example, there are four nodes,



we simply add 10 to each Path.

The original Pathway from s to y.

now is $sx + xy = 4$.

But after changing, it's 15 directly from s to y.

It ~~does~~ changes the shortest Path, which is different from the original.