Design & Analysis Of Algorithms

23K-0055 Assignment 4 BAI 5-A

(Q1) For a weighted directed graph $G = (V, E)$ w/o a negative-weight cycles, let m denote the maximum-over all vertices $v \in V$ of the smallest number of edges in any minimum weight path from s to v. The path relaxation property guarantees that after m iterations of the Bellman-Ford algorithm, every vertex v has already attained it's correct shortest-path distance $d[v]$. So in iteration $m+1$ no distance estimate will change.

1) Init $d[s] = 0$ & $d[v] = \infty$ for all other vertices

2) Repeat:

- Set flag changed = False
- for every edge $(u,v) \in E$, attempt to relax it
- if any relaxation succeeds, set changed = True

Until changed = False

test for a negative weight cycle has been removed. If there was a -ve weight cycle, the modified algo wouldn't get out of the loop because ~~done~~ some d value would change in each iteration

Date:

(Q2) (Dry run)

Initially: $d(r)=0, d(s)=\infty, d(t)=\infty, d(x)=\infty, d(y)=\infty, d(z)=\infty$

Process vertex r

$r \to s(5) \to d(s)=5$

$r \to t(3) \to d(t)=3$

Process vertex s

$s \to t(2) \to 5+2=7$ (no update, current 3)

$s \to x(6) \to d(x)=11$

Process vertex t

$t \to x(7) \to 3+7=10$ (update)

$t \to y(4) \to d(y)=7$

$t \to z(2) \to d(z)=5$

Process vertex x

$x \to y(-1) \to 10-1=9$ (no update)

$x \to z(1) \to 10+1=11$ (no update)

Process vertex y

$y \to z(-2) \to 7-2=5$ (no change)

Process vertex z

no outgoing edges

Date:

Final Shortest Path Distances from r

Vertex	Shortest Distance
r	0
s	5
t	2
x	6
y	5
z	3

$O(V+E)$ → time complexity.

V = vertices, E = edges

Initialization: $O(V)$, relaxing all edges once: $O(E)$

Q3) Initially:

$S = \phi$

$Q = \{s, x, y, z, t\}$ w/ distances: $d[z] = 0$, $d[s] = d[x] = d[y] = d[t] = \infty$

Iteration 1

Extract Min(Q): $u = z (d = 0)$

$S = \{z\}$

Relax edges from z:

$z \to x: d[x] = \min(\infty, 0+9) = 9$

$z \to s: d[s] = \min(\infty, 0+4) = 4$

$z \to t: d[t] = \min(\infty, 0+2) = 2$

Q state: $\{s(4), t(2), x(9), y(\infty)\}$

Date:

Iteration 2

Extract Min(Q): $u = t$ ($d = 2$)

$S = \{z, t\}$

Relax edges from t:

$t \to z$: $d[z] = 0$ (z already in S)

$t \to x$: $d[x] = \min(9, 2+6) = 8$

Q state: $\{s(4), x(8), y(\infty)\}$

Iteration 3

Extract-Min(Q): $u = s$ ($d = 4$)

$S = \{z, t, s\}$

Relax edges from s:

$s \to x$: $d[x] = \min(8, 4+10) = 8$ (no change)

$s \to y$: $d[y] = \min(\infty, 4+5) = 9$

Q state: $\{x(8), y(9)\}$

Iteration 4

Extract-Min(Q): $u = x$ ($d = 8$)

$S = \{z, t, s, x\}$

Relax edges from x:

$x \to t$: $d[t] = 2$ (t already in S)

$x \to z$: $d[z] = 0$ (z already in S)

Q state: $\{y(9)\}$

Date:

Iteration 5

Extract-Min(Q): u = y (d = 9)

S = {z, t, s, x, y}

Relax edges from y:

y → x : d[x] = 8 (x already in S)

y → z : d[z] = 0 (z already in S)

y → t : d[t] = 2 (t already in S)

Q-state: ∅

Final Result: d[z] = 0, d[s] = 4, d[t] = 2, d[x] = 8, d[y] = 9

(Source = z)

Date:

(4)

$$D^{(0)} \Rightarrow \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \pi^{(0)} \Rightarrow \begin{pmatrix} \phi & 1 & 1 & \phi & 1 \\ \phi & \phi & \phi & 2 & 2 \\ \phi & 3 & \phi & \phi & \phi \\ 4 & \phi & 4 & \phi & \phi \\ \phi & \phi & \phi & 5 & \phi \end{pmatrix}$$

$$D^{(1)} \Rightarrow \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \pi^{(1)} \Rightarrow \begin{pmatrix} \phi & 1 & 1 & \phi & 1 \\ \phi & \phi & \phi & 2 & 2 \\ \phi & 3 & \phi & \phi & \phi \\ 4 & 1 & 4 & \phi & 1 \\ \phi & \phi & \phi & 5 & \phi \end{pmatrix}$$

$$D^{(2)} \Rightarrow \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \pi^{(2)} \Rightarrow \begin{pmatrix} \phi & 1 & 1 & 2 & 1 \\ \phi & \phi & \phi & 2 & 2 \\ \phi & 3 & \phi & 2 & 2 \\ 4 & 1 & 4 & \phi & 1 \\ \phi & \phi & \phi & 5 & \phi \end{pmatrix}$$

$$D^{(3)} \Rightarrow \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \pi^{(3)} \Rightarrow \begin{pmatrix} \phi & 1 & 1 & 2 & 1 \\ \phi & \phi & \phi & 2 & 2 \\ \phi & 3 & \phi & 2 & 2 \\ 4 & 3 & 4 & \phi & 1 \\ \phi & \phi & \phi & 5 & \phi \end{pmatrix}$$

Date:

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \qquad \Pi^{(4)} = \begin{pmatrix} \phi & 1 & 4 & 2 & 1 \\ 4 & \phi & 4 & 2 & 1 \\ 4 & 3 & \phi & 2 & 1 \\ 4 & 3 & 4 & \phi & 1 \\ 4 & 3 & 4 & 5 & \phi \end{pmatrix}$$

$$D^{(5)} \Rightarrow \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \qquad \Pi^{(5)} \Rightarrow \begin{pmatrix} \phi & 3 & 4 & 5 & 1 \\ 4 & \phi & 4 & 2 & 1 \\ 4 & 3 & \phi & 2 & 1 \\ 4 & 3 & 4 & \phi & 1 \\ 4 & 3 & 4 & 5 & \phi \end{pmatrix}$$

Q5a)

Vertex	start (d)	finish (f)
B	1	18
C	2	13
F	3	12
H	4	11
G	5	8
D	6	7
I	9	10
E	14	17
A	15	16

B → C → F → H → G → D → I → E → A

Date:

Q5b)

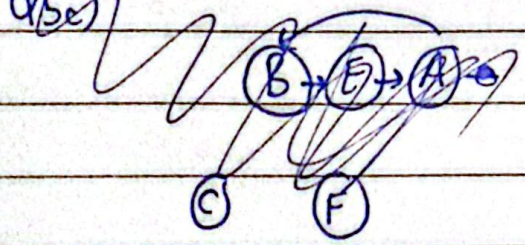
B → C	**Back edges**
B → E	A → B
C → F	D → H
F → H	I → H
H → G	
H → I	**forward edges**
G → D	F → I
E → A	

cross edges

A → D

E → H

Q5c)



Q5c) B → E → A → B

C, F } singleton

H → G → D → H , I → H

H → I

Q6a) BIPARTITE-CHECK (G):

1. Assign every vertex in the graph the color UNCOLORED
2. For each vertex s in the graph:

   If s is uncolored :

   - Assign s the color RED
   - Init empty Q
   - Add s to the queue

Date:

3. While q is not empty:
   Remove a vertex u from the queue
   For each adjacent vertex w of u:
      If w is uncolored:
         Assign w the opposite color of u
         Add w to the queue
      If w already has the same color as u:
         Report that the graph is not bipartite & terminate

4. If all vertices are processed w/o conflict:
   Report that graph is bipartite

Q6b) 2 colors

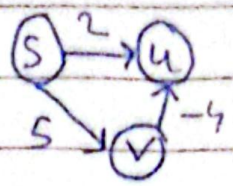
Q7a)

Step	Current Tree Nodes	Available Edges	Edge Selected	Weight
1	{A}	(A,G):12, (A,E):7, (A,B):5	(A,G)	12
2	{A,G}	(A,E):7, (A,B):5, (G,F):7, (G,E):4	(A,E) or (G,F)	7
3	{A,G,F}	(A,E):7, (A,B):5, (G,E):4, (F,C):2, (F,D):2, (F,E):3	(A,E)	7
4	{A,G,F,E}	(E,B):9, {E,C}:4, {A,B}:5, {F,C}:2, {F,D}:2	(E,B)	9
5	{A,G,F,E,B}	(B,C):7, (E,C):4, (F,C):2, (F,D):2	(B,C)	7
6	{A,G,F,E,B,C}	(C,D):5, (F,D):2	(C,D)	5

Date:

Q6b (Q7b) Yes, they can if graph has edges w/identical weights

Q7a)



$s \to v \to u = 5 + (-4) = 1$

Direct path $s \to u = 2$

is not shortest

Q7b) This method is not valid

Counter example:

Consider 2 paths from s to t

Path P_1:

1 edge

Original weight = 5

Path P_2:

2 edges

Original weight = 3

P_2 is shorter than P_1 ($3 < 5$)

Now adding $c = 10$ to each edge

$P_1 \Rightarrow 15$

$P_2 \Rightarrow 23$

Now P_1 appears shorter than P_2 so shortest path ordering has changed

Date:

Q7c) 1. In Dijkstra's algo, the source s is processed first.

2. All outgoing edges from s are relaxed before any other vertex is finalized

3. After this step, every vertex directly reachable from s has it's correct shortest distance

4. From this point onward, All remaining edges have +ve weights, so the standard correctness conditions of Dijkstra's algo apply.

5. Since no future relaxation can decrease a finalized distance using a -ve edge, no incorrect finalization occurs

If all negative-weight edges originate only from the source Dijkstra's algorithm remains correct.