HW1

Q1 Link Analysis -

Clearly if V_A, V_B, V_C & V_D are the proximity scores of all nodes in the network relative to the teleport sets A, B, C & D respectively; then we can do vector manipulations & find proximity scores for other users, with diff. teleport sets.

$A = \{1, 2, 3\}$
$B = \{3, 4, 5\}$
$C = \{1, 4, 5\}$
$D = \{1\}$

(this assumes that all teleports are equally likely, as given, & otherwise this argument breaks down)

1.1 $V_{Etoire} = V_{\{2\}} = V_A - (V_B + V_D - V_C) - V_D$

$$\boxed{V_{Etoire} = V_A - V_B + V_C - 2V_D}$$

1.2 We cannot calculate $V_{Felicity}$ (or $V_{\{5\}}$) with given information $\because$ we cannot isolate 5's contribution to proximity scores with this info.

1.3 ~~$\frac{1}{\beta}\{2V_A + V_B + V_C - V_D\}$~~

1.3 Yes we can do this &,

$$V_{Erlyinnis} = V_{\{1,2,3,4,5\}} = \left\{ \frac{1}{\beta}(2V_A + V_B + V_C) - V_D \right\} \times 0.1$$

(assumption- $W_A = [\beta, \beta, \beta, 0, 0]$
similarly for W_B & W_C
but $W_D = [1, 0, 0, 0, 0]$
$\because$ only 1 node)

1.4 ~~(this)~~ if we can isolate the linearly independent ~~(basis)~~ vectors from this set V; then we can calculate all ~~(vectors)~~ personalized pageRank vectors for users where teleport sets are a linear combination of these linearly independent subset of vectors. (A subspace)

1.5 $A = \beta M + \frac{(1-\beta)}{N} 1 1^T$

$$r = Ar \Rightarrow r = \left(\beta M + \frac{(1-\beta)}{N} 1 1^T\right) r$$

$$r = \beta M r + \frac{(1-\beta)}{N} 1 \underbrace{(1^T r)}$$

$\hookrightarrow$ dot product of 1 & r

$$\Rightarrow 1^T r = \sum_{i=1}^{N} r_i = 1$$

$$\therefore \boxed{r = \beta M r + \frac{1-\beta}{N} 1}$$

$$\therefore \boxed{r = \beta M r + \left(\frac{1-\beta}{N}\right) 1}$$

Ans 5.1 decoder is simply $z_1^T z_2$, i.e, $dec(z_1, z_2) = z_1^T z_2$

5.2 objective is $\min_{z} \|A - z^T W z\|_2$

I thought of W as an operator acting on z, then this follows.

5.3 eigen decomposition of A will be like

$$A = Q \Lambda Q^{-1}$$

$\because A$ is symmetric (Adjacency matrix)

$$\Rightarrow A = Q \Lambda Q^T$$

$$\min_{z} \|Q \Lambda Q^T - z^T W z\|_2$$

$\therefore W$ should be Λ (diagonal with eigenvalues of A) then ~~maybe~~ we will approximately get $\underline{\underline{z \approx Q^T}}$

5.4 the problem would become -

$$\min_{z} \|A^k - z^T z\|_2$$

$\because A^k$ has 1 in positions ~~of~~ where there is a path of length k between nodes.

struct2vec -

5.5 a new clique is formed with edges between 2 nodes having a set of weights $\{g_1(u,v), g_2(u,v), \ldots, g_k(u,v)\}$ which measures structural similarity upto dist. i (i.e., $g_i(u,v)$ is the similarity score b/w u & v considering their neighbourhoods to dist i)

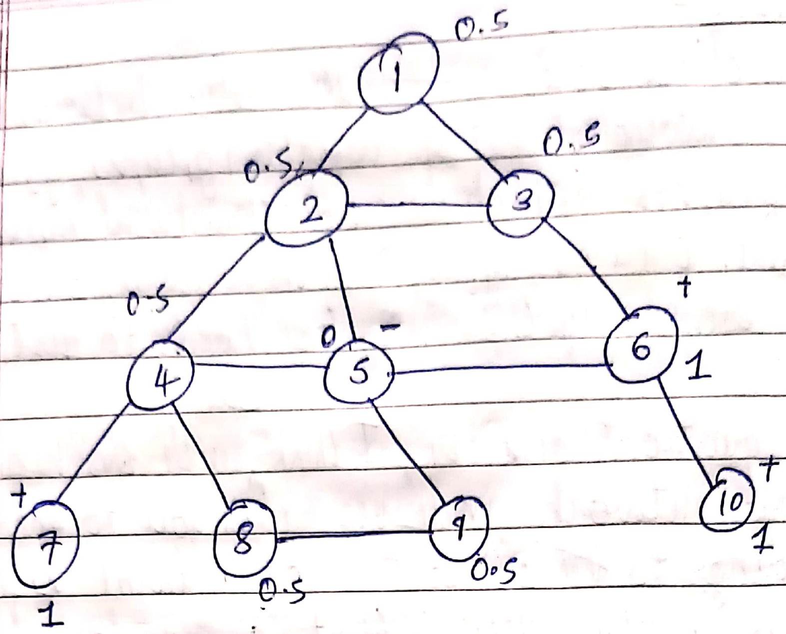
→ for the question's answer, I think even with this highly symmetric network node2vec will fail to encourage embeddings to ~~be st~~ capture structural info., ∵ it ~~has~~ is highly unlikely that a random walk from one clique will get to the other clique. ($V_{C_1}^T V_{C_2} \approx 0$)

5.6 with node2vec - we can reach all nodes in that specific clique (RHS one)
with struct2vec - we can reach all nodes in the network.

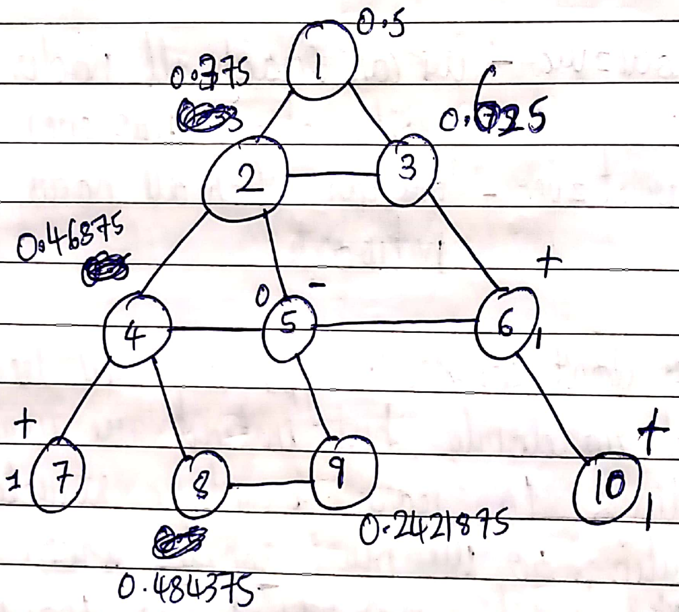
5.7 If we don't consider g_k's we will be walking around randomly but instead we want our embeddings to learn & capture structural similarity; so we must sample from a set of g_k's ~~(make it's probability high)~~ that have captured the similarity scores. In this way $P(v|z_u)$ will increase for u & v if they are structurally similar & our embeddings will grasp that.

5.8 no matter how we aggregate the 10 nodes in the 2 cliques after running struct2vec (Hadamard, add^n, avg., etc.) They will be extremely similar, i.e., $v_{C_1}^T \cdot v_{C_2} \approx 1$ (almost 1) ∵ they are very similar structurally.

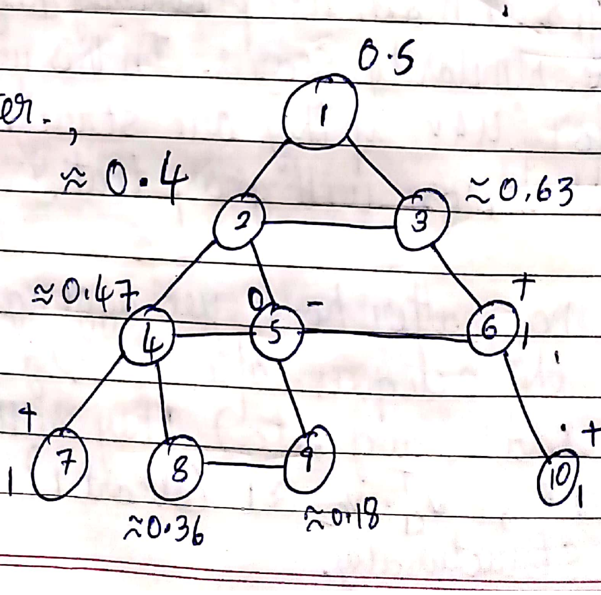
Q2



After 1st iteration,



after 2nd iter.,



$\therefore$ 2.1 $P(Y_3 = +) = 0.6328125$

2.2 $P(Y_4 = +) = 0.47073125$

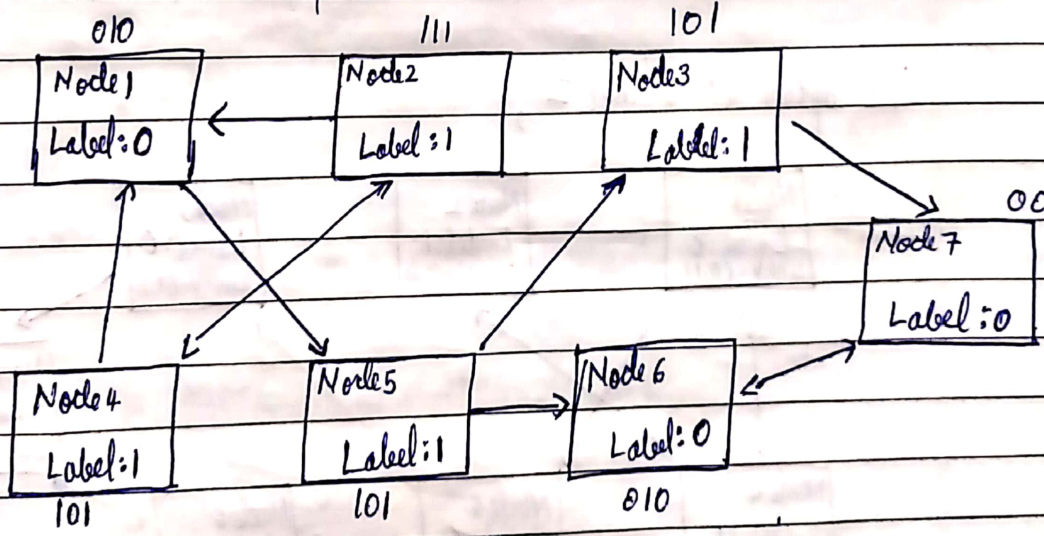
2.3 $P(Y_8 = +) = 0.35644653125$

2.4 1, 2, 4, 5, 8, 9 are "=" after 2nd iter.

Q3 $g(\text{WordV}, \text{LinkV}) = \begin{cases} 0 & I_0 = 1 \text{ or WordV} = 010 \\ 1 & \text{otherwise} \end{cases}$

WordV	f(WordV)
001	0
010	0
101	1
111	1

3.1



3.2 Links - { "Node 1": (0, 1),
"Node 2": (0, 1),
"Node 3": (0, 1),
"Node 4": (0, 1),
"Node 5": (1, 0),
"Node 6": (1, 1),
"Node 7": (1, 1) }

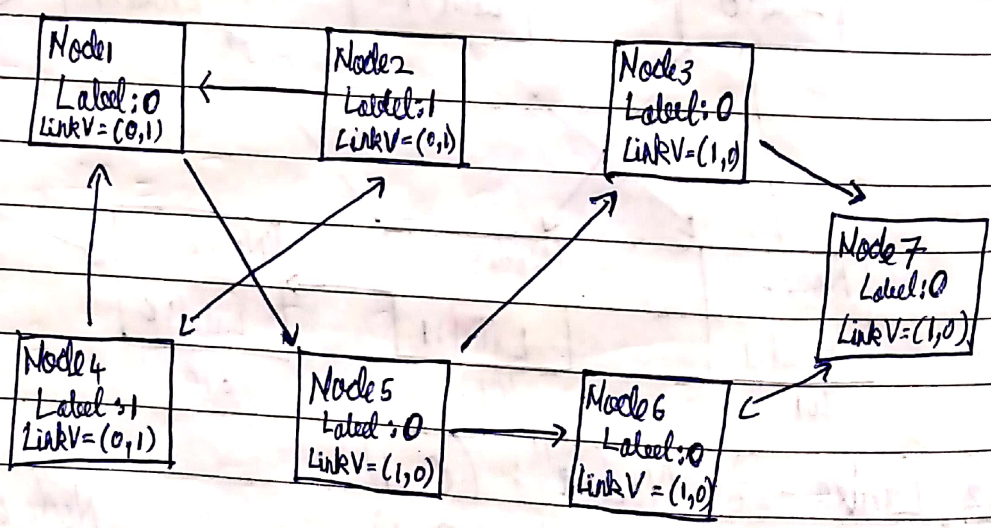
technically I_1 is useless but anyways.

3.2 Labels after iteration 1,

Labels = {"Node 1" : 0,
"Node 2" : 1,
"Node 3" : 1,
"Node 4" : 1,
"Node 5" : 0,
"Node 6" : 0,
"Node 7" : 0 }

LinkVs = {"Node 1" : (0,1),
"Node 2" : (0,1),
"Node 3" : (1,0),
"Node 4" : (0,1),
"Node 5" : (1,0),
"Node 6" : (1,0),
"Node 7" : (1,1) }

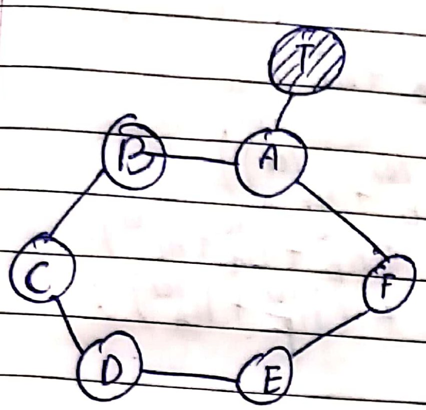
3.3 after iteration 2,



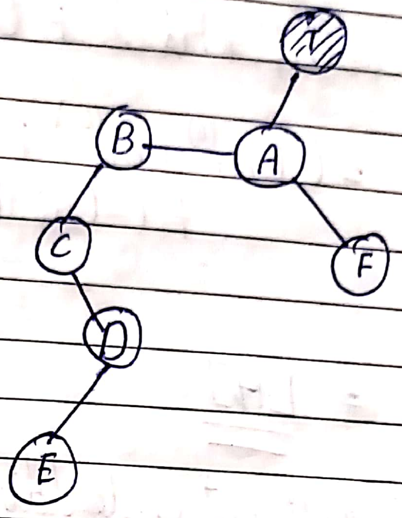
3.4 The labels have converged.

Q4 4.1 $h_v^{t+1} = h_v^{t} + \sum_{u \in N(v)} h_u^{t}$

LHS



RHS



~~Chop dg~~ ~~K hypoth~~ trying with K=2.

~~LHS~~

$\cancel{h_T^{(2)} = h_T^{(1)} + h_A^{(1)}}$.

LHS

RHS

$$h_T^{(2)} = h_T^{(1)} + h_A^{(1)}$$
$$= (h_T^{(0)} + h_A^{(0)}) + (h_A^{(0)} + h_B^{(0)} + h_F^{(0)})$$

] → clearly it will be same!

⇒ depth 2 is not sufficient.

K=3,

LHS

$$h_T^{(3)} = h_T^{(2)} + h_A^{(2)}$$
$$= (h_T^{(1)} + h_A^{(1)}) + (h_A^{(1)} + h_B^{(1)} + h_F^{(1)})$$
$$= ((h_T^{(0)} + h_A^{(0)}) + (h_T^{(0)} + h_A^{(0)} + h_B^{(0)} + h_F^{(0)})) + ((h_A^{(0)} + h_B^{(0)} + h_F^{(0)}) + (h_B^{(0)} + h_A^{(0)} + h_C^{(0)}) + (h_A^{(0)} + h_F^{(0)} + h_E^{(0)}))$$

$h_T^{(3)} = [15]$

RHS –

$$h_T^{(3)} = h_T^{(2)} + h_A^{(2)}$$

$$= (h_T^{(1)} + h_A^{(1)}) + (h_A^{(1)} + h_B^{(1)} + h_F^{(1)} + h_T^{(0)})$$

$$= ((h_T^{(0)} + h_A^{(0)}) + (h_A^{(0)} + h_B^{(0)} + h_F^{(0)}))$$
$$+ ((h_A^{(0)} + h_B^{(0)} + h_F^{(0)}) + (h_B^{(0)} + h_A^{(0)} + h_C^{(0)}) + (h_F^{(0)} + h_A^{(0)}))$$

$$= [13]$$

$\therefore$ different embeddings

this recurrence ~~a~~ approach is faster than drawing actual graphs & is equivalent to that.
We can imagine drawing ~~a~~ graphs with diff. brackets, if that even makes sense.

4.2 i) $r = \{0, 0, 1, 0\}$

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 0 & 1/3 \\ 1/2 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 \\ 1/2 & 1/2 & 1 & 0 \end{pmatrix} \end{array}$$

$\because$ if there is an edge from $i \to j$,
$$M_{ji} = \frac{1}{d_i}$$

ii) find x s.t.

$$Mx = x$$
$$(M-I)x = 0$$

$$\begin{pmatrix} -1 & 1/2 & 0 & 1/3 \\ 1/2 & -1 & 0 & 1/3 \\ 0 & 0 & -1 & 1/3 \\ 1/2 & 1/2 & 1 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$-a + \frac{b}{2} + \frac{d}{3} = 0 \rightarrow ①$$

$$\frac{a}{2} - b + \frac{d}{3} = 0 \rightarrow ②$$

$$-c + \frac{d}{3} = 0 \rightarrow ③ \Rightarrow \underline{\underline{c = d/3}}$$

$$\frac{a}{2} + \frac{b}{2} + c - d = 0 \rightarrow ④$$

$$\cancel{\frac{a}{2} = b + \frac{d}{3}}$$

① − ②,
$$-\frac{3a}{2} + \frac{3b}{2} = 0 \Rightarrow \underline{\underline{a = b}}$$

from ④, $a + c - d = 0$
$$a = d - c$$
$$= 2d/3$$

$\therefore a = 2d/3, \; b = 2d/3, \; c = d/3, \; d = d$

$$\cancel{\text{in ①, } -\frac{2d}{3} + \frac{d}{3} + \frac{d}{3}}$$

$$\Rightarrow x = d\,(2/3 \quad 2/3 \quad 1/3 \quad 1)^T$$

$x^T x = 1$
$$\Rightarrow d^2\left(\frac{4}{9} + \frac{4}{9} + \frac{1}{9} + 1\right) = 1 \quad \Rightarrow \underline{\underline{d = 1/\sqrt{2}}}$$

$$\therefore x = [\sqrt{2}/3 \quad \sqrt{2}/3 \quad 1/3\sqrt{2} \quad 1/\sqrt{2}]$$

$r = [0.47, 0.47, 0.24, 0.71]$

4.3 ~~T, ~~

transition matrix, $T = D^{-1}A$

in the slides as well,

but intuition -

$$D = \begin{pmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{pmatrix}$$

$$D^{-1} = \begin{pmatrix} 1/d_1 & & & 0 \\ & 1/d_2 & & \\ & & \ddots & \\ 0 & & & 1/d_n \end{pmatrix}$$

i.e., D^{-1} has the normalization factors for each node.

$$(T = M^T)$$

4.4 $$h_i^{(l+1)} = \frac{1}{2} h_i^{(l)} + \frac{1}{2} \times \frac{1}{|N_i|} \sum_{j \in N_i} h_j^{(l)}$$

$$T = \frac{1}{2} D^{-1}A + \frac{1}{2} H^{(l)}$$

$$T = \frac{1}{2}\left(D^{-1}A + H^{(l)}\right)$$

4.4 $$h_i^{(l+1)} = \frac{1}{2} h_i^{(l)} + \frac{1}{2} \times \frac{1}{|N_i|} \sum_{j \in N_i} h_j^{(l)}$$

$$\Rightarrow T = \frac{1}{2} D^{-1}(I+A) = \frac{1}{2} D^{-1} + \frac{1}{2} D^{-1} A$$

much better intuition than

~~this is like adding a self loop at each vertex with importance = normal edges~~

this is like adding a self loop at each vertex with importance = Σ(normal out edges from the vertex).

4.5 $$H^{(l)} = (D^{-1}A) H^{(l-1)}$$

$H^{(l)} = (D^{-1}A)^l H^{(0)}$; informally -

$\Rightarrow$ effects of nodes keep getting diluted.

if by chance $H_i^{(k)} \approx c \quad \forall i$ then

$$H_i^{(k+1)} = \frac{1}{\cancel{|N_i|}} \times c \cancel{|N_i|} = c \quad \forall i$$

$\Rightarrow$ then it will become constant.

formally, with the given hint, this can be considered as a Markov Chain with $D^{-1}A$ as the transition matrix also ...

<u>Irreducibility</u> - $(D^{-1}A)$ is irreducible $\because$ it can take us from any state to any other state.

4.6 message f^n is just <u>passing on the embedding</u>.

aggregation f^n is <u>just sum</u> of messages from neighbours + itself.

update step is $\begin{cases} 1 & \text{if aggr.} > 0 \\ 0 & \text{otherwise} \end{cases}$

~~i.e., $H^{(l+1)} =$ for $k = 1 \dots K$ do for $v \in V$ do $h^k_{N(v)}$ ... $H^{(l+1)} = (I+A)$~~

$$h_i^{(l)} = \sum_{u \in N(i)} h_u^{(l-1)} + h_i^{(l-1)}$$

$$h_i^{(l)} = \mathbb{I}\{ h_i^{(l)} > 0 \}$$

$$\cancel{H^{(l)} = \mathbb{1}\{(I+A)H^{(l-1)} > 0\}}$$

$$H^{(l)} = \mathbb{1}\{(I+A)H^{(l-1)} > 0\}$$

↳ indicator f^n (elementwise checks)