CS 132 – Spring 2020, Assignment 9

Answer A

$$A^k = PD^kP^{-1} = \begin{bmatrix} 3 & 4 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2^k & 0 \\ 0 & 1^k \end{bmatrix} \times \begin{bmatrix} -1 & 4 \\ 1 & -3 \end{bmatrix} = \begin{bmatrix} 3 \times 2^k & 4 \times 1^k \\ 2^k & 1^k \end{bmatrix} \times \begin{bmatrix} -1 & 4 \\ 1 & -3 \end{bmatrix} = \begin{bmatrix} -3 \times 2^k + 4 & -12 + 12 \times 2^k \\ -2^k & -3 + 4 \times 2^k \end{bmatrix}$$

$$\boxed{\begin{bmatrix} -3 \times 2^k + 4 & -12 + 12 \times 2^k \\ -2^k & -3 + 4 \times 2^k \end{bmatrix}}$$

Answer B

As per the Diagonalization Theorem, the eigenvalues of A are the diagonal entries of D, and the eigenvectors are the respective columns of P, so:

$$\boxed{\lambda = 5 | \lambda = 4}$$

$$\boxed{\lambda = 5 : \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}$$

$$\boxed{\lambda = 4 : \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}}$$

Answer C

Since the matrix is triangular, with identical diagonal entries, then the only possible eigenvalue is 5, so for $\lambda = 5$:

$$A - 5I = 0 \equiv \begin{bmatrix} 5 & 1 \\ 0 & 5 \end{bmatrix} - \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 - free \\ x_2 = 0 \end{cases}$$

$$\boxed{\text{Since the general solution is } \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ we cannot get a basis for } R^2\text{, so the matrix is not diagonalizable.}}$$

Answer D

First thing we need to do is to find the eigenvalues of A, to do so we need to find the characteristic equation:

$$det(A - \lambda I) = 0 \equiv det \begin{bmatrix} 2-\lambda & 3 \\ 4 & 1-\lambda \end{bmatrix} = 0 \equiv (2-\lambda)(1-\lambda) - 12 = 0$$
$$\lambda^2 - 3\lambda - 10 = 0 \equiv \lambda = -2 | \lambda = 5$$

Now that we know the eigenvalues, we can calculate the vectors, so for $\lambda = -2$:

$$A + 2I = 0 \equiv \begin{bmatrix} 4 & 3 & 0 \\ 4 & 3 & 0 \end{bmatrix} \equiv \begin{bmatrix} 4 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 = -\frac{3}{4}x_2 \\ x_2 - free \end{cases} \equiv \begin{bmatrix} -\frac{3}{4} \\ 1 \end{bmatrix}$$

Now for $\lambda = 5$:

$$A - 5I = 0 \equiv \begin{bmatrix} -3 & 3 & 0 \\ 4 & -4 & 0 \end{bmatrix} \equiv \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 = x_2 \\ x_2 - free \end{cases} \equiv \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\boxed{\text{So, we we have that } P = \begin{bmatrix} -\frac{3}{4} & 1 \\ 1 & 1 \end{bmatrix} \text{ and } D = \begin{bmatrix} -2 & 0 \\ 0 & 5 \end{bmatrix}}$$

Answer E

Since we already know the eigenvalues for the matrix ($\lambda = 2, 8$), then we can go straight to calculate their vectors, so, for $\lambda = 2$:

$$A - 2I = 0 \equiv \begin{bmatrix} 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 = -x_2 - x_3 \\ x_2 - free \\ x_3 - free \end{cases} \equiv \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Now for $\lambda = 8$:

$$A - 8I = 0 \equiv \begin{bmatrix} -4 & 2 & 2 & 0 \\ 2 & -4 & 2 & 0 \\ 2 & 2 & -4 & 0 \end{bmatrix} \equiv \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 = 0 \\ x_2 = -x_3 \\ x_3 - free \end{cases} \equiv \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

$$\boxed{\text{So, we we have that } P = \begin{bmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \text{ and } D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix}}$$

Answer F

a. True, as by the Theorem 6 7, for a matrix to be diagonalizable, then it must either have n distinct eigenvalues, hence generating n eigenvectors. Or the sum of the multiplicity of its eigenvalues must be equal to n, hence generating n eigenvectors.

b. False, as per Theorem 7, a can have non-distinct eigenvalues, as long as the sum of the dimensions of the eigenspaces equals to n

c. True, as per Theorem 5

d. True, as per the IMT

Answer G

No, A is not diagonalizable, as it doesn't fulfill any of the requirements:
A doesn't have n distinct eigenvalues.
The sum of the dimensions of the eigenspaces of A does not equal n.

Answer H

a.
The transition matrix is:

$$\boxed{\begin{bmatrix} 0 & 1/3 & 0 & 1/2 & 1/2 \\ 1/3 & 0 & 1/2 & 0 & 1/2 \\ 0 & 1/3 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/2 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 \end{bmatrix}}$$

To find the steady state vector, we have that $Ax = x \equiv Ax - x = 0 \equiv (A - I)x = 0$:

$$\begin{bmatrix} -1 & 1/3 & 0 & 1/2 & 1/2 & 0 \\ 1/3 & -1 & 1/2 & 0 & 1/2 & 0 \\ 0 & 1/3 & -1 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1/2 & -1 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & -1 & 0 \end{bmatrix} \equiv \begin{bmatrix} -1 & 1/3 & 0 & 1/2 & 1/2 & 0 \\ 0 & 1 & -9/16 & -3/16 & -3/4 & 0 \\ 0 & 0 & 1 & -9/13 & -4/13 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 = \frac{3}{2}x_5 \\ x_2 = \frac{3}{2}x_5 \\ x_3 = x_5 \\ x_4 = x_5 \\ x_5 - free \end{cases} \equiv \boxed{\begin{bmatrix} \frac{3}{2} \\ \frac{3}{2} \\ 1 \\ 1 \\ 1 \end{bmatrix}}$$

b.

The transition matrix is:

$$\boxed{\begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \end{bmatrix}}$$

To find the steady state vector, we have that $Ax = x \equiv Ax - x = 0 \equiv (A - I)x = 0$:

$$\begin{bmatrix} -1 & 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & -1 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & -1 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 & -1 & 0 \end{bmatrix} \equiv \begin{bmatrix} 1 & -1/3 & -1/3 & -1/3 & 0 \\ 0 & 1 & -1/2 & -1/2 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 = x_4 \\ x_2 = x_4 \\ x_3 = x_4 \\ x_4 - free \end{cases} \equiv \boxed{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}$$

Answer I

```
alpha = 0.01

P1 = np.array([[0, 0, 1, 0, 0],
               [1/3, 0, 0, 1/2, 0],
               [1/3, 0, 0, 1/2, 0],
               [1/3, 1/2, 0, 0, 0],
               [0, 1/2, 0, 0, 0]])

getRanking(P1, alpha, "P1")
```

```
P1

Transition Matrix:
 [[0.    0.    1.    0.    0.   ]
 [0.333 0.    0.    0.5   0.   ]
 [0.333 0.    0.    0.5   0.   ]
 [0.333 0.5   0.    0.    0.   ]
 [0.    0.5   0.    0.    0.   ]]

Google Matrix:
 [[0.002 0.002 0.992 0.002 0.2  ]
 [0.332 0.002 0.002 0.497 0.2  ]
 [0.332 0.002 0.002 0.497 0.2  ]
 [0.332 0.497 0.002 0.002 0.2  ]
 [0.002 0.497 0.002 0.002 0.2  ]]

PageRank:
 [0.521 0.463 0.463 0.463 0.291]

Node Order:
 [1 4 3 2 5]
```

```
P2 = np.array([[0, 1/2, 1/4, 0, 0, 0],
               [0, 0, 1/4, 0, 0, 0],
               [0, 1/2, 0, 1/2, 0, 0],
               [0, 0, 1/4, 0, 1/2, 0],
               [0, 0, 1/4, 1/2, 0, 0],
               [0, 0, 0, 0, 1/2, 0]])

getRanking(P2, alpha, "P2")
```

```
P2

Transition Matrix:
 [[0.   0.5  0.25 0.   0.   0.  ]
 [0.   0.   0.25 0.   0.   0.  ]
 [0.   0.5  0.   0.5  0.   0.  ]
 [0.   0.   0.25 0.   0.5  0.  ]
 [0.   0.   0.25 0.5  0.   0.  ]
 [0.   0.   0.   0.   0.5  0.  ]]

Google Matrix:
 [[0.167 0.497 0.249 0.002 0.002 0.167]
 [0.167 0.002 0.249 0.002 0.002 0.167]
 [0.167 0.497 0.002 0.497 0.002 0.167]
 [0.167 0.002 0.249 0.002 0.497 0.167]
 [0.167 0.002 0.249 0.497 0.002 0.167]
 [0.167 0.002 0.002 0.002 0.497 0.167]]

PageRank:
 [0.478 0.477 0.477 0.36  0.359 0.241]

Node Order:
 [3 4 5 1 6 2]
```

```
import numpy as np

def remCol0(A):
    B = A
    n = len(B)

    for j in range(len(B[0])):
        b = True
        for k in range(n):
            if (B[k][j] != 0):
                b = False
        if b:
            for k in range(n):
                B[k][j] = 1/n

    return B

def stageTwo(A, a):
    B = A
    x = (1 - a)
    y = a / (len(B))

    for i in range(len(B)):
        for j in range(len(B[0])):
            B[i][j] = x * B[i][j] + y

    return B

def getRanking(P, a, name):
    print(name)
    print()
    print("Transition Matrix: \n", np.round(P,3))

    P1 = remCol0(P)
    P2 = stageTwo(P1, a)

    values, vectors = np.linalg.eig(P2)
    indices = np.argsort(values)
    principal = indices[-1]
    steadyState = np.real(vectors[:,principal])
    reverseOrder = np.argsort(steadyState)
    order = 1 + reverseOrder[::-1]

    print()
    print("Google Matrix: \n", np.round(P2,3))
    print()
    print("PageRank: \n", np.round(steadyState[order - 1],3))
    print()
    print("Node Order: \n", order)
```

Answer J

a.

If we re-write this we get that $A = \begin{bmatrix} p_0 & (1-p_1) \\ (1-p_0) & p_1 \end{bmatrix}$

Now to find the eigenvalues we must first find the characteristic equation so:

$$det(A-\lambda I) \equiv det\begin{bmatrix} p_0-\lambda & (1-p_1) \\ (1-p_0) & p_1-\lambda \end{bmatrix} \equiv (p_0-\lambda)(p_1-\lambda)-(1-p_1)(1-p_2) \equiv$$

$$\equiv (p_0 \times p_1) - (p_1\lambda) - (p_0\lambda) + \lambda^2 - 1 + p_1 + p_2 - (p_1 \times p_2) \equiv$$

$$\equiv \lambda^2 + (p_0+p_1)(1-\lambda) - 1 = 0$$

If we solve this we get: $\lambda = p_0 + p_1 - 1 | \lambda = 1$, since we know that both p_0 and p_1 are smaller than 1, then we know the biggest eigenvalue will ALWAYS be 1.

b.

Know that we know the eigenvalue of 1, we can figure out the vector with $(A-I)x = 0$:

$$\begin{bmatrix} (p_0-1) & (1-p_1) & 0 \\ (1-p_0) & (p_1-1) & 0 \end{bmatrix} \equiv \begin{bmatrix} -p_0 & (1-p_1) & 0 \\ (1-p_0) & -p_1 & 0 \end{bmatrix} \equiv \begin{bmatrix} -p_0 & (1-p_1) & 0 \\ 0 & 0 & 0 \end{bmatrix} \equiv \begin{cases} x_1 = \frac{(1-p_1)}{p_0}x_2 \\ x_2 - free \end{cases} \equiv$$

$$\boxed{\begin{bmatrix} \frac{(1-p_1)}{p_0} \\ 1 \end{bmatrix}}$$

c.

To find this, all we have to do is make the eigenvector we last calculated be a steady state vector, and then replace the values of p_1 and p_2. However for simplicity sake I'll be doing it backwards, first replacing values and then adapting the vector:

$$\begin{cases} x_1 = 0.0316x_2 \\ x_2 - free \end{cases},$$

$$x_1 + x_2 = 1 \equiv 1.0316x_2 = 1 \equiv \begin{cases} x_1 = 0.0306 \\ x_2 = 0.9694 \end{cases}$$

$\boxed{\text{On average there will be 969 packets in the network.}}$