SDSC3001 Tutorial 3

Matrix Multiplication

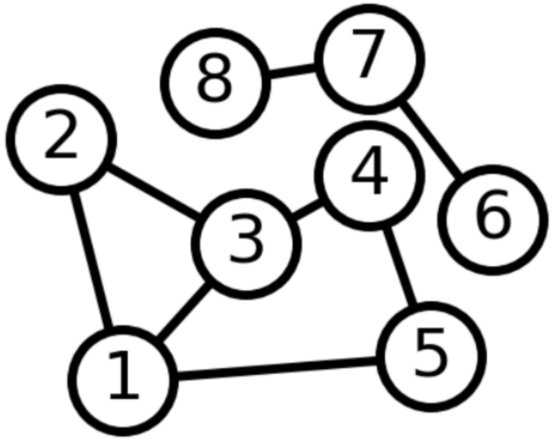
2023.09.25

Outline

- Review of Graph Data
- Matrix Multiplication and Representation of Sparse Matrix
- Power Iteration of PageRank

Review of Graph Data

Undirected Graph



8×8 matrix

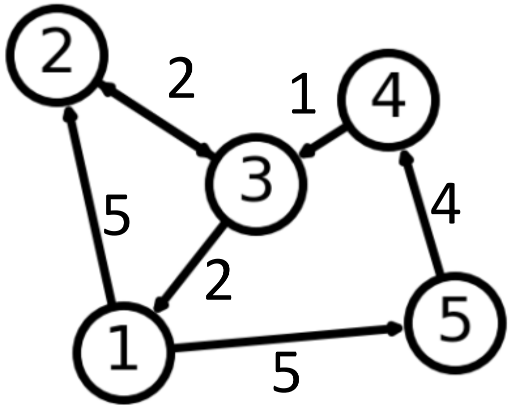
$$A_{8,8} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$a_{ij} = 1$, *if node* i, j *are connected*

$a_{ij} = 0$, *otherwise*

Review of Graph Data

Directed Graph



No edge from node 2 to node 1

An edge with weight of 5 from node 1 to node 2

$$A_{5,5} = \begin{bmatrix} 0 & 5 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \end{bmatrix}$$

Time complexity

Matrix-vector multiplication

- A matrix of shape (n, n) multiplied by a vector of shape $(n, 1)$

$$y = Ax = \begin{bmatrix} 0 & 2 & 1 \\ 4 & 0 & 3 \\ 1 & 4 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \times 2 + 2 \times 1 + 1 \times 0 \\ 4 \times 2 + 0 \times 1 + 3 \times 0 \\ 1 \times 2 + 4 \times 1 + 0 \times 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 6 \end{bmatrix}$$

- $n \times n = O(n^2)$ multiplications

Time complexity

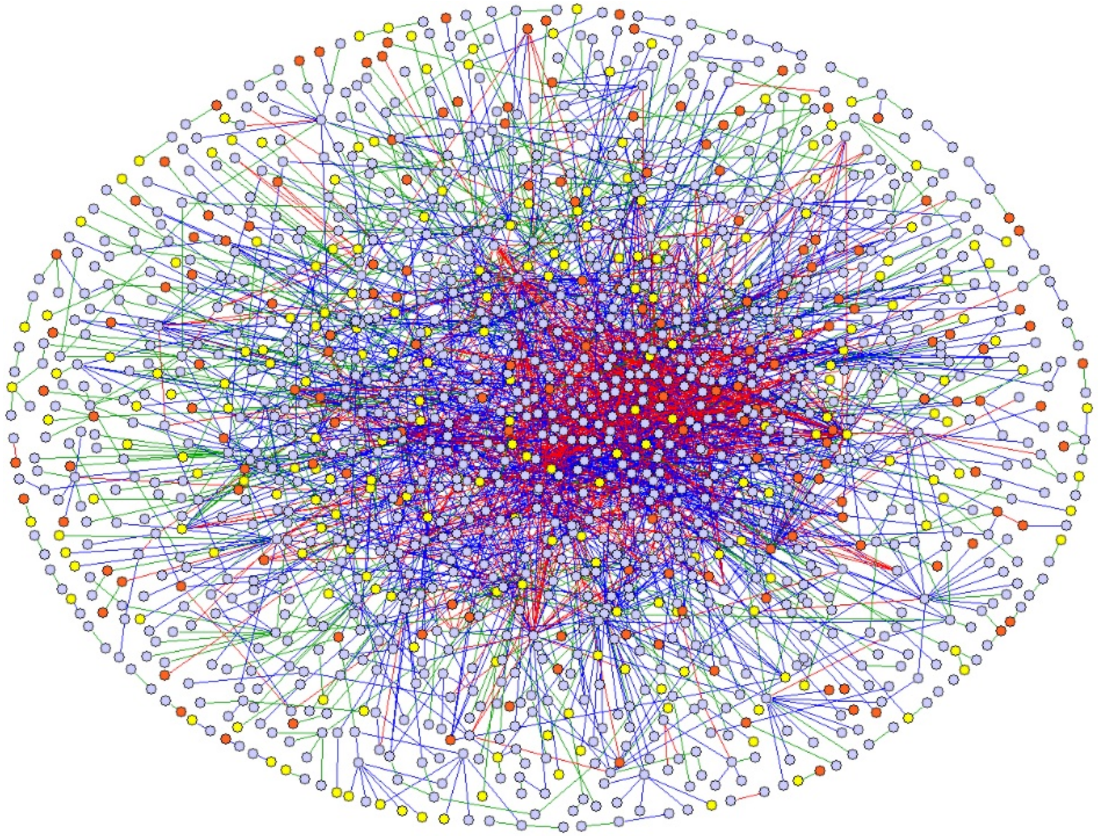
Matrix-matrix multiplication

- Two matrices of shape (n, n)

$$y = AB = \begin{bmatrix} 0 & 2 & 1 \\ 4 & 0 & 3 \\ 1 & 4 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 2 & 4 \\ 3 & 0 & 0 \\ 1 & 5 & 0 \end{bmatrix} = \underbrace{A \cdot \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}}_{O(n^2)} + A \cdot \begin{bmatrix} 2 \\ 0 \\ 5 \end{bmatrix} + A \cdot \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}$$

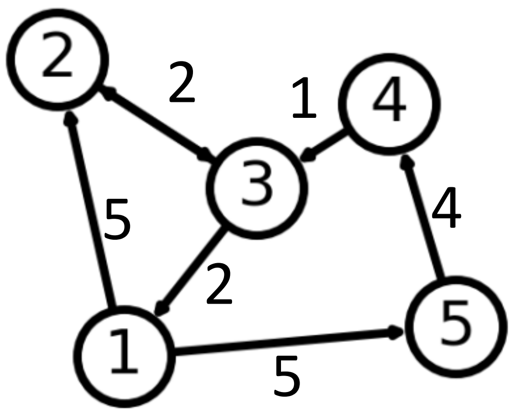
- $n^2 \times n = O(n^3)$ multiplications

Real-world Large Graph



n is large!

Sparse Matrix

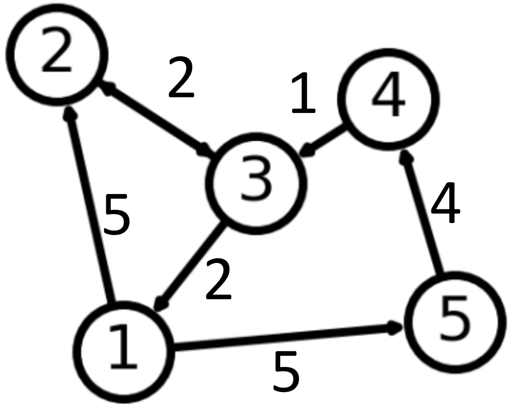


$$A_{5,5} = \begin{bmatrix} 0 & 5 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \end{bmatrix}$$

- Space complexity is $O(n^2)$
- Only a few number of elements are not zero

Coordinate list (COO)

Let m be the number of non-zero values.



$$\begin{array}{rcc} \mathit{index}: & \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} & \\ A_{5,5} = & \begin{bmatrix} 0 & 5 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

- V = [5 5 2 2 2 1 4]
- COL_INDEX = [1 4 2 0 1 2 3]
- ROW_INDEX = [0 0 1 2 2 3 4]

The space complexity is $O(3m)$.

Compressed sparse row (CSR)

Let m be the number of non-zero values and n be the number of rows.

$$\begin{array}{ccc} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} & \\ A = & \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \end{array}$$

$index$: 0 1 2 3 4 5

The space complexity is $O(2m + n + 1)$.

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

Explanation 1

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

Compressed sparse row (CSR)

Let m be the number of non-zero values and n be the number of rows.

$$\begin{array}{cc} & \begin{array}{cccccc} index: & 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ A = & \begin{bmatrix} \boxed{\begin{matrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \end{matrix}} \\ \begin{matrix} 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{matrix} \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \end{array}$$

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

Value k=4 and the index i=2

⇒ in the first 2 rows there are 4 non-zero elements (10, 20, 30, 40).

Compressed sparse row (CSR)

Let m be the number of non-zero values and n be the number of rows.

$$\begin{array}{rcc} & \begin{matrix} \mathit{index}: & 0 & 1 & 2 & 3 & 4 & 5 \end{matrix} & \\ A = & \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} & \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \end{array}$$

The space complexity is $O(2m + n + 1)$.

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

Explanation 2

Another explanation: Element (except the last element) with (index, value) = (i, k) represents that the first non-zero element in row i is V[k].

Compressed sparse row (CSR)

Let m be the number of non-zero values and n be the number of rows.

$$\begin{array}{cc} & \begin{array}{cccccc} \text{index}: 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \end{array}$$

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

Value k=4 and the index i=2 ⇒ V[4]=50.

The first non-zero element in row 2 is 50.

CSR Two Problems

- 1. Given a sparse matrix, how to define (V, COL_INDEX, ROW_INDEX)?
- 2. Given (V, COL_INDEX, ROW_INDEX), how to rebuild the sparse matrix/ find the location of a certain value?

1. Given a sparse matrix, how to define (V, COL_INDEX, ROW_INDEX)?

$$
\begin{array}{cc}
\textit{index}: & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} & \\
A = & \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array}
\end{array}
$$

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]

ROW_INDEX = [ ]

Explanation 1

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

Explanation 2

Element (except the last element) with (index, value) = (i, k) represents that the first non-zero element in row i is V$[k]$.

Q_1: how many elements are there in “ROW_INDEX ”?

i:0~4, the length of “ROW_INDEX ” is **5**

ROW_INDEX = [ ]

1. Given a sparse matrix, how to define (V, COL_INDEX, ROW_INDEX)?

$$\begin{array}{ccccccccc} index: & 0 & 1 & 2 & 3 & 4 & 5 & \\ & & & & & & & \\ A = & \left[\begin{matrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{matrix}\right] & \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \end{array}$$

Explanation 1

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

k is obtained by counting the number

- V = [10 20 30 40 50 60 70 80]

***i*=0**

In the first 0 rows (DO NOT EXIST), there are 0 non-zero elements

ALWAYS

ROW_INDEX = [0]

1. Given a sparse matrix, how to define (V, COL_INDEX, ROW_INDEX)?

$$\begin{array}{c|c} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ A = & \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \end{array}$$

index: 0 1 2 3 4 5

Explanation 1

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

k is obtained by counting the number

- V = [10 20 30 40 50 60 70 80]

In the first 1 rows, there are 2 non-zero elements

***i*=1**

ROW_INDEX = [0 | 2 | | 7 |]

***i*=3**

In the first 3 rows, there are 7 non-zero elements

1. Given a sparse matrix, how to define (V, COL_INDEX, ROW_INDEX)?

$$\begin{array}{ccccccc} \textit{index}: & 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array}$$

- V = [10 20 30 40 50 60 70 80]

Idx_V: 0 1 2 3 4 5 6 7

ROW_INDEX = [0]

Explanation 2

Element (except the last element) with (index, value) = (i, k) represents that the first non-zero element in row i is V[k].

k is obtained by finding index in V

***i*=0**

first non-zero element in row **0** is “10”

Its idx in V is “0”

K=0

1. Given a sparse matrix, how to define (V, COL_INDEX, ROW_INDEX)?

$$\begin{array}{c|cccccc|c} index: & 0 & 1 & 2 & 3 & 4 & 5 & \\ \hline & & & & & & & \end{array}$$

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array}$$

- V = [10 20 30 40 50 60 70 80]

Idx_V: 0 1 2 3 4 5 6 7

Explanation 2

Element (except the last element) with (index, value) = (i, k) represents that the first non-zero element in row i is V[k].

k is obtained by finding index in V

***i*=2**

first non-zero element in row **2** is "50"

V(4)=50; "50" idx in V is "4"

K=4

ROW_INDEX = [0 2 4]

Compressed sparse row (CSR)

Let m be the number of non-zero values and n be the number of rows.

$$\begin{array}{cc} & \begin{array}{cccccc} \color{red}{0} & \color{red}{1} & \color{red}{2} & \color{red}{3} & \color{red}{4} & \color{red}{5} \end{array} \\ \color{red}{index:} & \end{array}$$

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix} \begin{array}{c} \color{red}{0} \\ \color{red}{1} \\ \color{red}{2} \\ \color{red}{3} \end{array}$$

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

Point to the first non-zero element in each row

2.Given (V, COL_INDEX, ROW_INDEX), how to rebuild the sparse matrix/ find the location of a certain value?

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix}$$

Explanation 1

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

Explanation 2

Element (except the last element) with (index, value) = (i, k) represents that the first non-zero element in row i is V[k].

Q_1: Size of Matrix A?

Length(ROW_INDEX)-1 * max(COL_INDEX)+1

2.Given (V, COL_INDEX, ROW_INDEX), how to rebuild the sparse matrix/ find the location of a certain value?

Explanation 1

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

***i*=0**

ROW_INDEX(0)=0 always

***i*=1** ROW_INDEX(1)=2

In the first 1 rows, there are 2 non-zero elements (10&20)

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.Given (V, COL_INDEX, ROW_INDEX), how to rebuild the sparse matrix/ find the location of a certain value?

Explanation 1

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 2 4 7 8]

***i*=2** ROW_INDEX(2)=4

In the first 2 rows, there are 4 non-zero elements (10 20 30 40)

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 00 & 40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.Given (V, COL_INDEX, ROW_INDEX), how to rebuild the sparse matrix/ find the location of a certain value?

- V = [10 20 30 40 50 60 70 80]
- COL_INDEX = [0 1 1 3 2 3 4 5]
- ROW_INDEX = [0 $\boxed{2\ 4}$ 7 8]

Explanation 1

Element with (index, value) = (i, k) represents that there are k non-zero elements in the first i rows.

How to get the coordinates of the element "40" (Column=?, Row=?) ?

1. Find idx in V: k = V.index(40)=3

2. Find the col: COL_INDEX.index(k)=3

3. Find the row: ROW_INDEX[**1**]=2 $\leq k <$ ROW_INDEX[**2**]=4 $\Rightarrow$ Row = 1

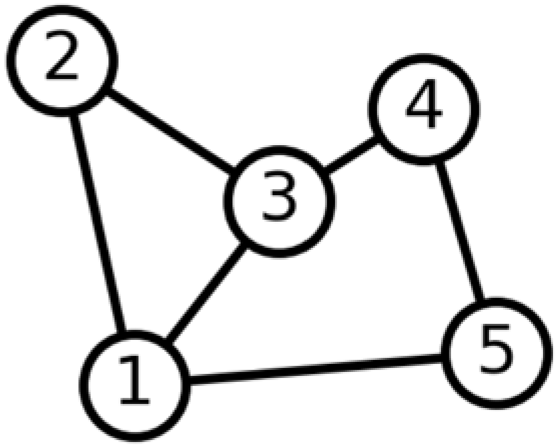
So the coordinate of element 40 is (1, 3)

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix}$$

Coding: PageRank

PageRank (**PR**) :

- an algorithm used by Google Search to rank web pages in their search engine results
- used to represent the likelihood that a person randomly clicking on links will arrive at any particular page.



Adjacency Matrix

$$A_{5,5} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Degree Matrix

$$D = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

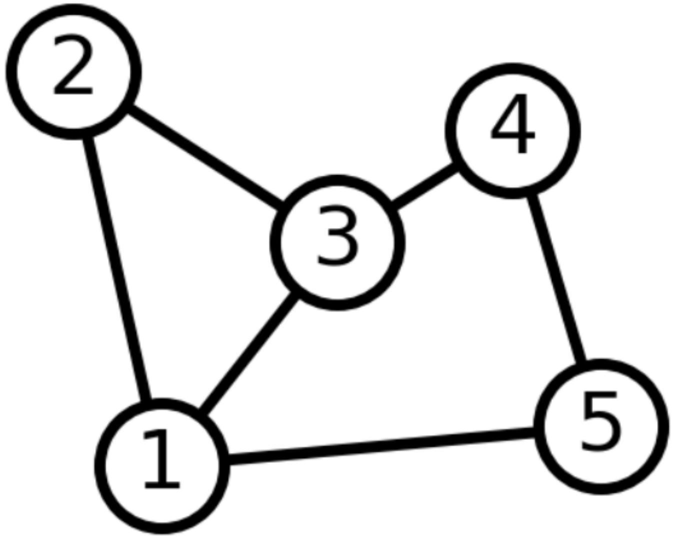
* $\deg(v_i)$of a vertex counts the number of times an edge terminates at that vertex.

Coding: PageRank

- Pr(1->2) = Pr(1->3) = Pr(1->5) = 1/3
- Pr(2>1) = Pr(2->3) = 1/2

$$P = D^{-1}A$$

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



transition probability

PageRank

- Power iteration method

starting with an arbitrary vector

- $\boldsymbol{\pi}(0) = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})^\top$

operator

the operator is applied in succession

- $\boldsymbol{\pi}(t) \leftarrow \boxed{\left((1-\alpha)\mathbf{P}^\top + \frac{\alpha}{n}\mathbf{E}\right)} \boldsymbol{\pi}(t-1)$
- Repeat iterations until $\|\boldsymbol{\pi}(t) - \boldsymbol{\pi}(t-1)\| \leq \epsilon$

$$\boxed{(1-\alpha)P^T\pi(t-1)} + \frac{\alpha}{n}E\pi(t-1)$$

Sparse **Matrix-vector multiplication**

π(): probability distribution
P: transition probability matrix
E: matrix with all ones

Using CSR for Sparse
Matrix-vector multiplication

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{bmatrix}$$

$$x = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6]^T$$

```python
# CSR format matrix A
v = [10, 20, 30, 40, 50, 60, 70, 80]
COL_INDEX = [0,  1,  1,  3,  2,  3,  4,  5]
ROW_INDEX = [0,  2,  4,  7,  8]

# vector
x = [1, 2, 3, 4, 5, 6]

# y = Ax
y = [0, 0, 0, 0]
m = ROW_INDEX[-1]  # the number of non-zero values


'''
In each row, figure out two things:
- 1. what are the non-zero elements?
= 2. their corresponding col idx.
'''

for i in range(len(ROW_INDEX) - 1):
    row_index_start = ROW_INDEX[i]
    row_index_end = ROW_INDEX[i+1]

    for j in range(row_index_start, row_index_end):
        # Length of the range list is the number of valid value in each row.
        # v[j]: non-zero element
        # COL_INDEX[j]: corresponding col idx

        col = COL_INDEX[j]
        y[i] += v[j] * x[col]

print(y)
```

```
[50, 220, 740, 480]
```

- Power iteration method
  - $\boldsymbol{\pi}(0) = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})^\top$
  - $\boldsymbol{\pi}(t) \leftarrow \left((1-\alpha)\mathbf{P}^\top + \frac{\alpha}{n}\mathbf{E}\right)\boldsymbol{\pi}(t-1)$
  - Repeat iterations until $\|\boldsymbol{\pi}(t) - \boldsymbol{\pi}(t-1)\| \leq \epsilon$

$$\boxed{(1-\alpha)P^T\pi(t-1)} + \frac{\alpha}{n}E\pi(t-1)$$

Sparse **Matrix-vector multiplication**

```python
num_nodes = len(ROW_INDEX) - 1
pi = np.ones(num_nodes) / num_nodes
print(pi)

alpha = 0.01
second_term = alpha * np.ones(num_nodes) / num_nodes
while True:
    new_pi = np.zeros(num_nodes)
    # (1-alpha)P * pi + alpha 1/n
    for i in range(len(ROW_INDEX) - 1):
        row_index_start = ROW_INDEX[i]
        row_index_end = ROW_INDEX[i + 1]

        for j in range(row_index_start, row_index_end):
            col = COL_INDEX[j]
            value = v[j]
            new_pi[i] += value * pi[col]
    new_pi = new_pi * (1-alpha)+ second_term* pi

    # check convergence by L1-distance(a, b) = sum_i |a_i - b_i|
    if np.sum(np.abs(new_pi - pi)) < 1e-4:
        break
    else:
        pi = new_pi

print(pi)

f = open("pi.pickle", 'wb')
pickle.dump(pi, f)
```

$\boldsymbol{\pi}(0)$

Matrix-vector multiplication

$P^T\pi(t-1)$

termination condition

```
[0.5 0.5 0.03225806 ... 0.14285714 0.14285714 0.14285714]
[ 6547 15408 17793 ... 216687 256538 260898]
[ 0 2 33 ... 2312483 2312490 2312497]
[3.54731947e-06 3.54731947e-06 3.54731947e-06 ... 3.54731947e-06
```

Thank you!