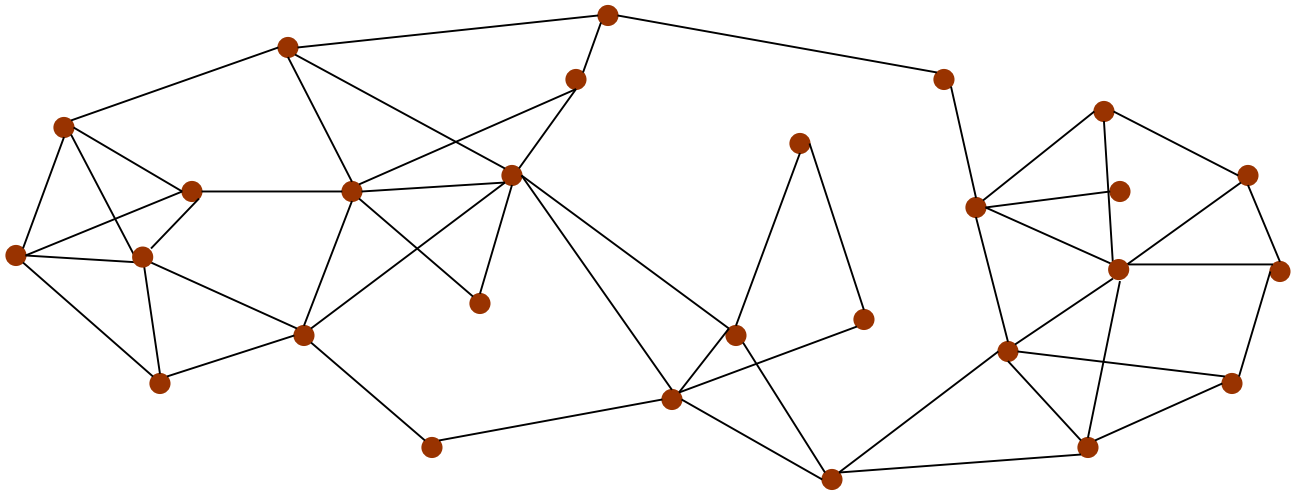
Data Structure
Programming Project #1

郭建志

Background

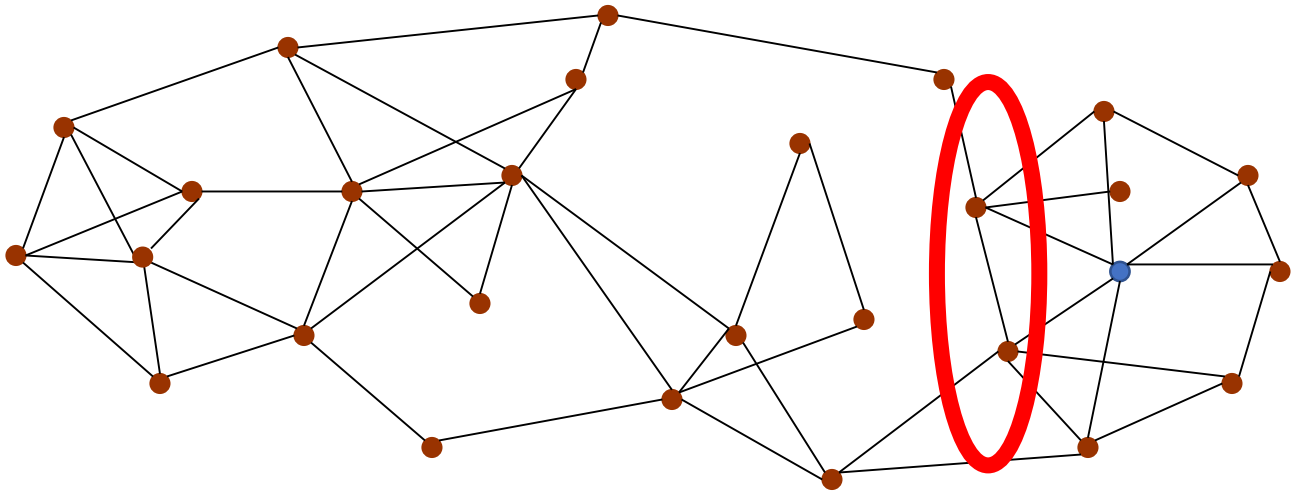
- Large-scale sensor networks
- Every node collect the environment information
- How to obtain the average data?
  Such as temperature



Background

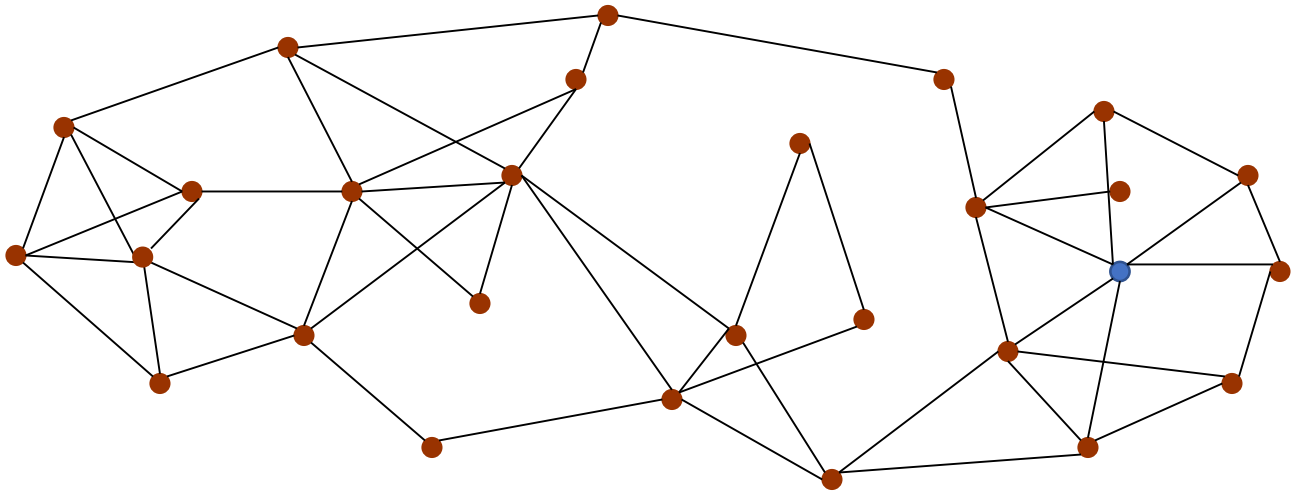
- A sink averages all data
- The nodes near the sink may send more packets

→ exhaust energy → node failure
→ network partition



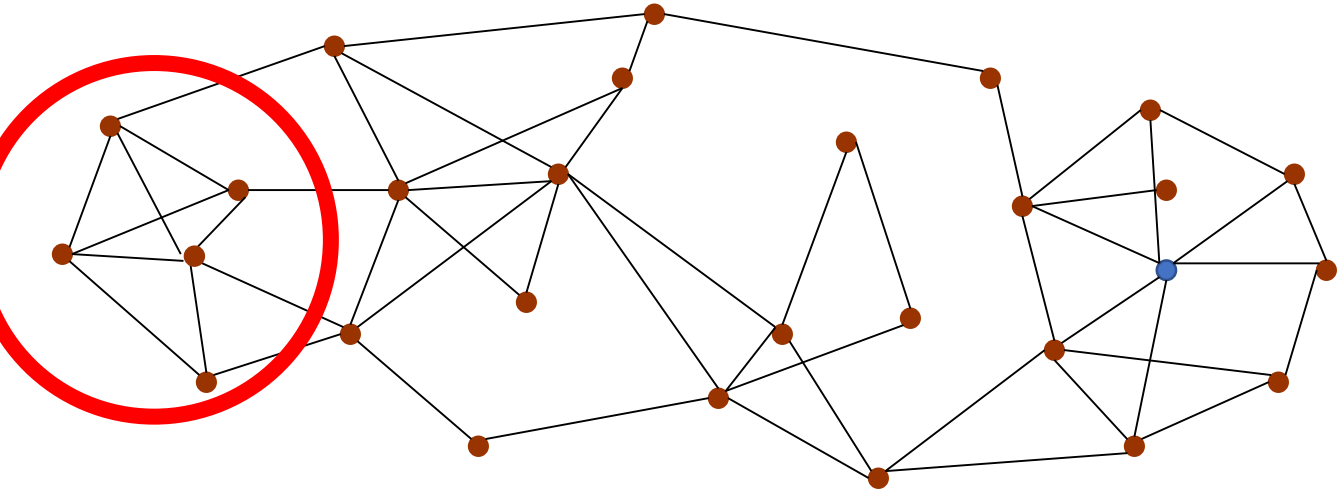
Background

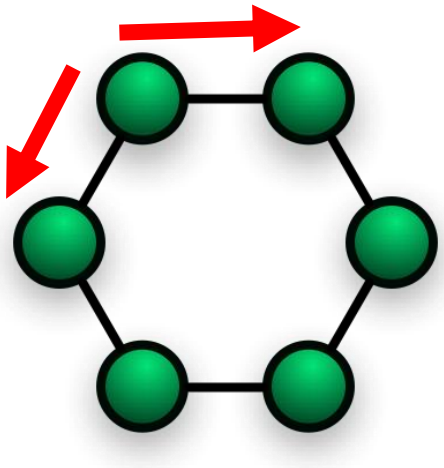
- A sink averages all data
- Centralized algorithms are not suitable for large-scale networks
- How will you design a distributed algorithm to obtain the average?



Distributed Consensus Algorithm

- Every node acts a sensor and a sink at the same time
- At each iteration,
- each node sends the data to its neighbors
- each node aggregate the data from its neighbors



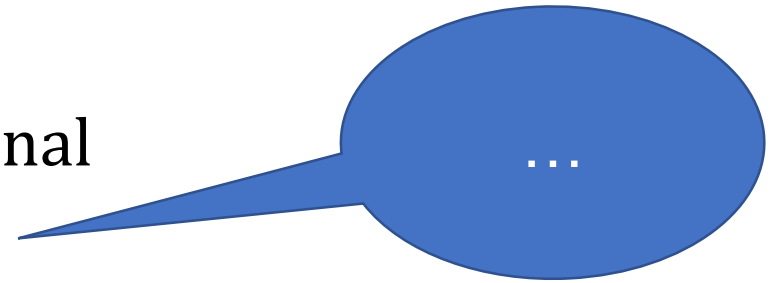


Programming Project #1: Convergence Matrix

- Input:
  - Number of nodes
  - Nodes with non-negative coordinates (x, y) (the input graph is connected when we add links if dist(u,v) < 1)
- Procedure:
  - Add a link between any two nodes u, v as dist(u,v) < 1
  - Average the data from the neighbors
- Output:
  - The data of nodes at each iteration
- The grade is inversely proportional to the number of iterations for convergence

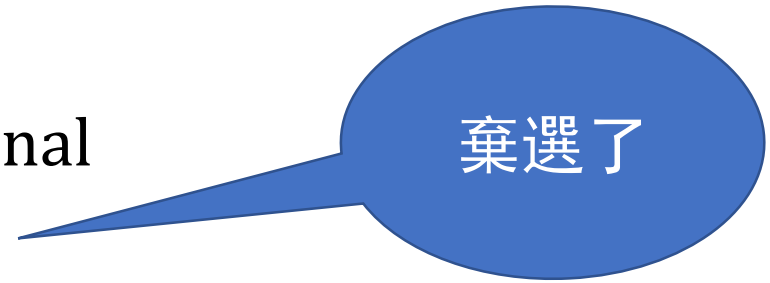
Programming Project #1: Convergence Matrix

- Input:
  - Number of nodes
  - Nodes with non-negative coordinates (x, y) (the input graph is connected when we add links if dist(u,v) < 1)
- Procedure:
  - Add a link between any two nodes u, v as dist(u,v) < 1
  - Average the data from the neighbors
- Output:
  - The data of nodes at each iteration
- The grade is inversely proportional to the number of iterations for convergence



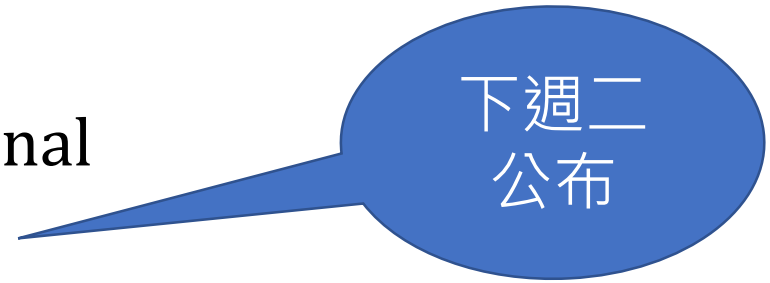
Programming Project #1: Convergence Matrix

- Input:
  - Number of nodes
  - Nodes with non-negative coordinates (x, y) (the input graph is connected when we add links if dist(u,v) < 1)
- Procedure:
  - Add a link between any two nodes u, v as dist(u,v) < 1
  - Average the data from the neighbors
- Output:
  - The data of nodes at each iteration
- The grade is inversely proportional to the number of iterations for convergence



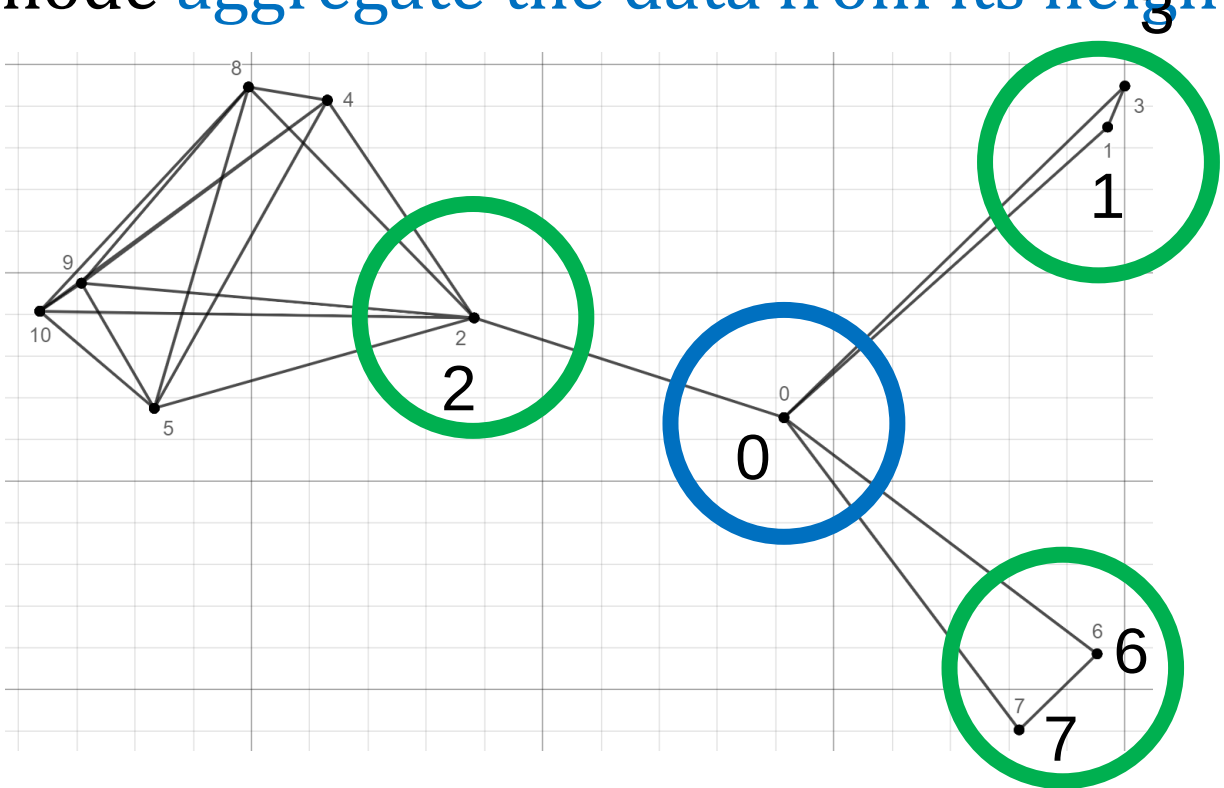
Programming Project #1: Convergence Matrix

- Input:
  - Number of nodes
  - Nodes with non-negative coordinates (x, y) (the input graph is connected when we add links if dist(u,v) < 1)
- Procedure:
  - Add a link between any two nodes u, v as dist(u,v) < 1
  - Average the data from the neighbors
- Output:
  - The data of nodes at each iteration
- The grade is inversely proportional to the number of iterations for convergence



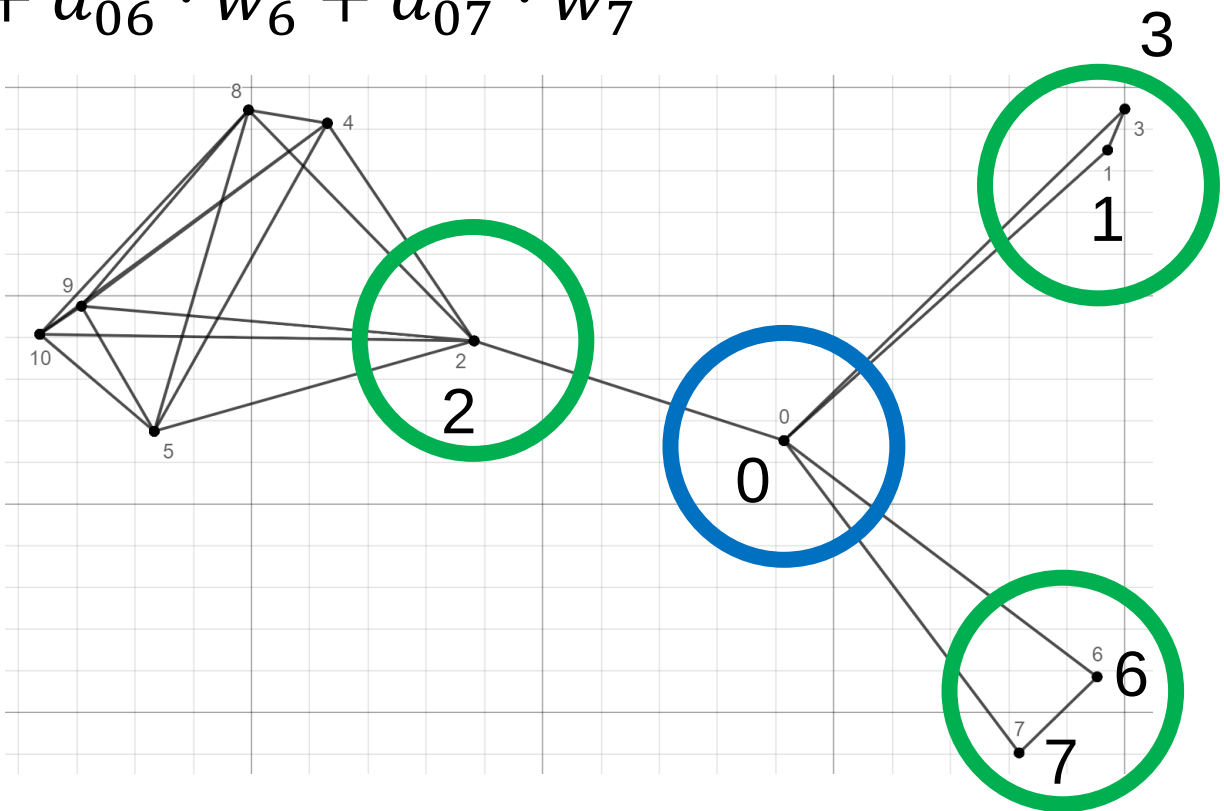
Programming Project #1: Convergence Matrix

- At each iteration,
- each node sends the data to its neighbors
- each node aggregate the data from its neighbors



Programming Project #1: Convergence Matrix

- $w_i' = \sum_{j \in V} a_{ij} \cdot w_i$
- e.g., $w_0' = a_{00} \cdot w_0 + a_{01} \cdot w_1 + a_{02} \cdot w_2 + a_{03} \cdot w_3 + a_{06} \cdot w_6 + a_{07} \cdot w_7$



Review

- $\begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$
- Eigenvector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ with eigenvalue 1
- Eigenvector $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ with eigenvalue 0.4
- $Ax = \lambda x$

- Doubly stochastic matrix
- $w_1 = Aw_0 = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \end{bmatrix}$
- $w_2 = Aw_1 = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \end{bmatrix} = \begin{bmatrix} 5.8 \\ 4.2 \end{bmatrix}$
- …
- $w_0 = \begin{bmatrix} 10 \\ 0 \end{bmatrix} = 5 \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 5 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$
- $w_n = A^n w_0 = 5(1)^n \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 5(0.4)^n \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

When n is ∞ ➔ 0

Convergence Matrix

- Lazy-Metropolis-based consensus matrix

- $$a_{ij} = \begin{cases} 1 - \sum_{k \in V \setminus \{i\}} a_{ik}, & \text{if } i = j \\ \frac{1}{2 \cdot \max\{\deg(j), \deg(i)\}}, & \text{else if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Input Sample

#nodes #rounds

nodeID x y value

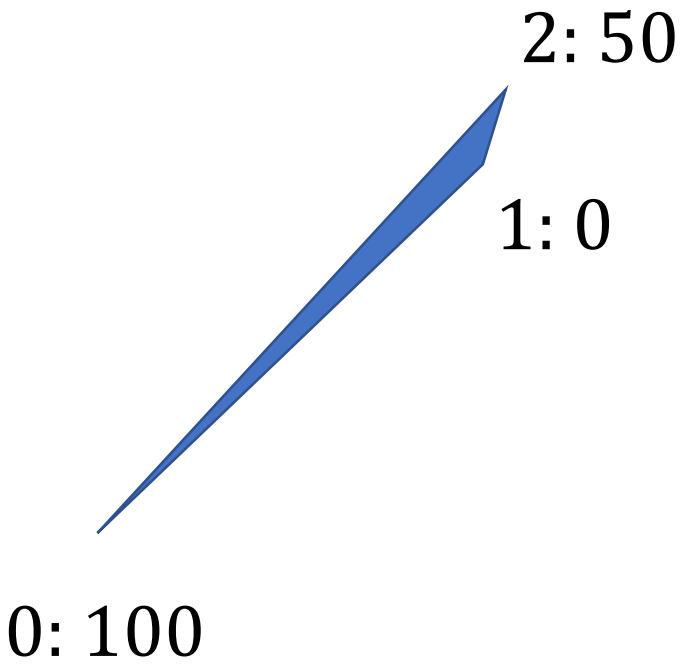
...

3 3

0 1.4142 1.1534 100

1 1.97 1.85 0

2 1.9996 1.9484 50

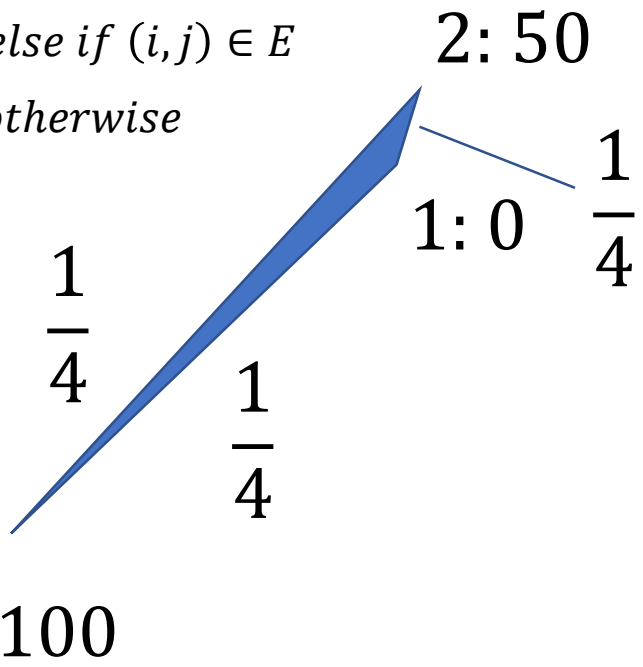


Out Sample

- $w_{ij} = \begin{cases} 1 - \sum_{k \in V \setminus \{i\}} w_{ik}, & if\ i = j \\ \frac{1}{2 \cdot \max\{\deg(j), \deg(i)\}}, & else\ if\ (i,j) \in E \\ 0, & otherwise \end{cases}$

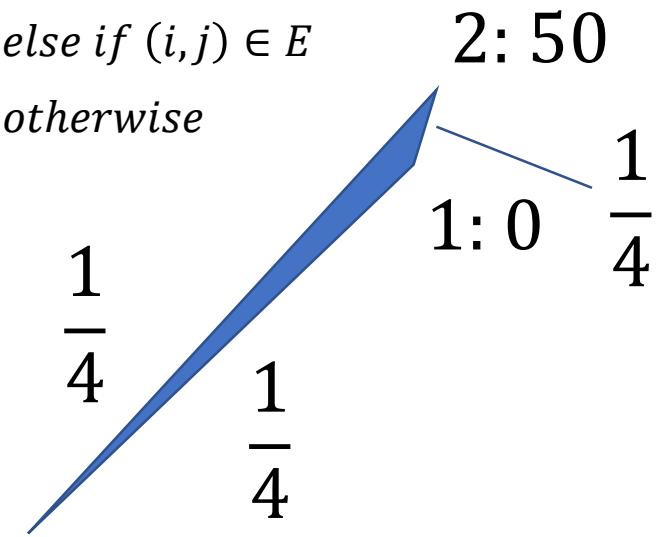
3 3

100 0 50



Out Sample

- $w_{ij} = \begin{cases} 1 - \sum_{k \in V \setminus \{i\}} w_{ik}, & if\ i = j \\ \frac{1}{2 \cdot \max\{\deg(j), \deg(i)\}}, & else\ if\ (i,j) \in E \\ 0, & otherwise \end{cases}$



3 3

100	0	50
62.5	37.5	50

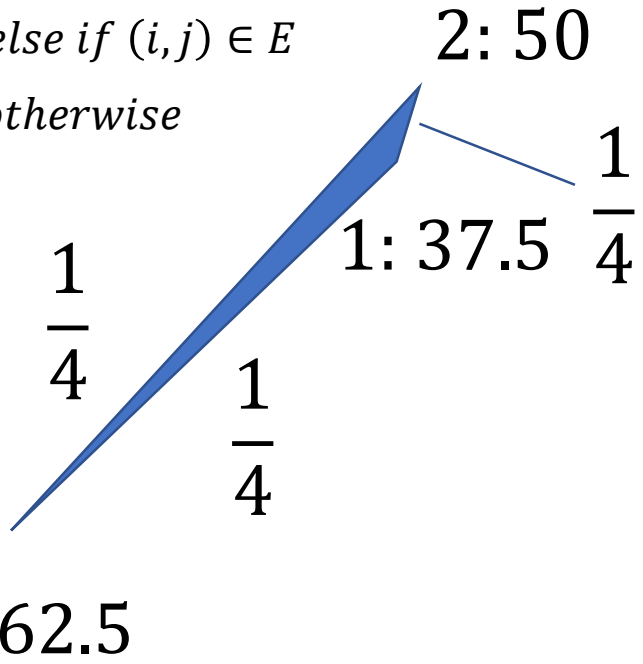
$0'\colon\ 100 \cdot \frac{1}{2} + 50 \cdot \frac{1}{4} = 62.5$

$1'\colon 50 \cdot \frac{1}{4} + 100 \cdot \frac{1}{4} = 37.5$

$2'\colon 50 \cdot \frac{1}{2} + 100 \cdot \frac{1}{4} = 50$

Out Sample

- $w_{ij} = \begin{cases} 1 - \sum_{k \in V \setminus \{i\}} w_{ik}, & if\ i = j \\ \frac{1}{2 \cdot \max\{\deg(j), \deg(i)\}}, & else\ if\ (i,j) \in E \\ 0, & otherwise \end{cases}$



3	3	
100	0	50
62.5	37.5	50
53.125	46.875	50

$$0': 62.5 \cdot \frac{1}{2} + 50 \cdot \frac{1}{4} + 37.5 \cdot \frac{1}{4} = 53.125$$

$$1': 37.5 \cdot \frac{1}{2} + 50 \cdot \frac{1}{4} + 62.5 \cdot \frac{1}{4} = 46.875$$

$$2': 50 \cdot \frac{1}{2} + 62.5 \cdot \frac{1}{4} + 37.5 \cdot \frac{1}{4} = 50$$

#nodes #rounds
value1 value2 value3…
…

Out Sample

3 3
100.00 0.00 50.00
62.50 37.50 50.00
53.125 46.875 50.00

Note that the precision is set to 2
注意每個數字僅需列印出小數點下兩位
不需要另外處理四捨五入 (使用%.2f 輸出即可)

Further reading

- Eigenvalue and eigenvector
- Metropolis-based consensus matrix
- Doubly stochastic matrix

Note

- Superb deadline: 9/29 Tue
- Deadline: 10/6 Tue
- Submit your code to E-course2
- Demonstrate your code in 工院1館 401B
- **C Source code**
- Show a good programming style