

## 1. Chuyển từ hh:mi:ss:ms → ms và ngược lại

### 1.1. Phương pháp chuyển

- hh → ms = hh x 60 x 60 x 1000
- mi → ms = mi x 60 x 1000
- ss → ms = ss x 1000
- Kết quả = hh → ms + mi → ms + ss → ms + ms
- ms = 10.000 ticks

### 1.2. Mẹo tính máy tính CasioFX – 570VN PLUS

- Chuyển từ fomate → ms bằng cách tạo một biểu thức tổng quát rồi ấn CALC để nhập giá trị biến của biểu thức rồi ấn =
  - Ví dụ đổi hh:mi:ss.ms → ms: thì tạo biểu thức:
    - $A \times 60 \times 60 \times 1000 + B \times 60 \times 1000 + C$  (C là ss.ms nhưng bỏ dấu . nhập liền nhau)
  - Lưu kết quả tính toán vào các biến khác trong máy tính (các biến này phải khác các biến đã dùng trong biểu thức (dùng phím SHIFT + STO + Tên biến cần lưu).
- Chuyển từ ms → fomate thì dùng chia lấy dư dùng SHIFT + —
  - Ví dụ đổi ms → hh:mi:ss.ms
    - ms :R(60 x 60 x 1000) = thương, R = số dư → hh = thương
    - Lấy số dư :R(60 x 1000) = thương, R = số dư → mi = thương
    - ms = 3 số cuối của số dư, các số đầu còn lại là ss

## 2. Giải thuật Schlosser

### 2.1. Đề bài:

- Có N nút mạng.

### 2.2. Cách làm:

- Số lượng láng giềng cho mỗi nút  $m = \log_2(N)$  (làm tròn lên)
- Viết các nút vào bảng dưới đây từ 0 ... (N-1) và chuyển nó sang dạng nhị phân

Nút	1 (0...0)	2 (0...1)	...	...	...	...	...	...	N - 1 (1...1)
1 (0...0)									
2 (0...1)									
...									
...									
...									
...									
N - 1 (1...1)									

- **Thực hiện điền các ô vào bảng như sau:**
  - Xét từng nút k trong mỗi hàng (nút có m bit)
    - **for i from 0 to m-1** (lặp từ phải qua trái, bit cuối cùng là 0, bit đầu tiên là m-1)
      - Thay đổi bit i trong nút k (đổi tạm thời, không lưu lại, cuối mỗi lần lặp thì mã bit của nút k lại trở về ban đầu).
        - Nếu bit i là 0 thì đổi thành 1
        - Nếu bit i là 1 thì đổi thành 0
      - Gán mã bit vừa thay đổi vào mã bit h
      - Điền giá trị của i+1 vào ô (trong cùng một hàng) mà mã bit tại cột của ô đó bằng với mã bit h.
  - **Chú ý:**
    - Với mỗi giá trị N nhất định thì chỉ có một kết quả điền vào bảng

- Bảng có tính đối xứng qua đường chéo chính
- **Quá trình truyền thông điệp tại một nút  $k$  bất kỳ (Nút  $k$  ở hệ cơ số 10):**
  - **Quá trình truyền thông điệp tại một nút  $k$  bất kỳ với ngưỡng là  $a$ :**
    - Xem nút  $k$  nằm ở hàng nào trong bảng kia. Giả sử nằm ở hàng  $i$
    - Các nút mà  $k$  chuyển thông điệp cho là các giá trị nút của các cột  $j$  mà tại đó ô  $(i, j)$  có giá trị  $> a$ .
    - Nếu nút  $j$  được nút  $k$  chuyển tới có giá trị nhận tại ô  $(i, j) < m$  thì nút  $j$  tiếp tục chuyển cho các nút khác với ngưỡng là giá trị nhận tại ô  $(i, j)$ .
  - **Quá trình truyền thông điệp tại một nút  $k$  bất kỳ:**
    - Là quá trình truyền thông điệp tại một nút  $k$  bất kỳ với ngưỡng là  $a = 0$ .
    - Số thông điệp cần chuyển:  $2^m - 1$
- **Ví dụ minh họa với  $N = 8$** 
  - Số lượng láng giềng cho mỗi nút  $m = \log_2(8) = 3$

Nút	0 (000)	1 (001)	2 (010)	3 (011)	4 (100)	5 (101)	6 (110)	7 (111)
0 (000)		1	2		3			
1 (001)	1			2		3		
2 (010)	2			1			3	
3 (011)		2	1					3
4 (110)	3					1	2	
5 (101)		3			1			2
6 (110)			3		2			1
7 (111)				3		2	1	

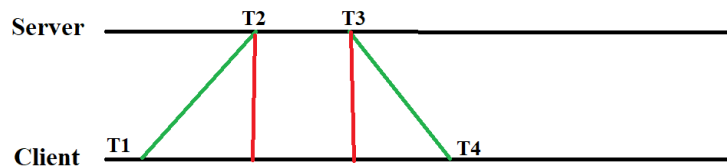
- Quá trình truyền thông điệp tại nút 0
  - Nút 0 chuyển cho nút 1 (nhãn  $1 < m$ ), 2 (nhãn  $2 < m$ ), 4 (nhãn  $3 == m$ )  $\rightarrow$  (3 thông điệp)
    - Nút 1 (nhãn 1) chuyển cho nút 3 (nhãn  $2 < m$ ), 5 (nhãn  $3 == m$ )  $\rightarrow$  (2 thông điệp)
      - Nút 3 (nhãn 2) chuyển cho nút 7 (nhãn  $3 == m$ )  $\rightarrow$  (1 thông điệp)
    - Nút 2 (nhãn 2) chuyển cho nút 6 (nhãn  $3 == m$ )  $\rightarrow$  (1 thông điệp).
  - Số thông điệp cần chuyển:  $3 + 2 + 1 + 1 = 7 = 2^3 - 1$

### 2.3. Mẹo

- Học thuộc đối với điền bảng
- Suy ra đường đi của thông điệp tại nút bất kỳ thì tự suy

### 3. Giải thuật Cristian

#### 3.1. Đề bài



- Máy khách thực hiện đồng bộ thời gian vật lý với máy chủ, thời gian gửi và nhận các thông điệp như sau:
  - Thời gian máy khách gửi ( $T_1$ )
  - Thời gian máy chủ nhận ( $T_2$ )
  - Thời gian máy chủ gửi ( $T_3$ )
  - Thời gian máy chủ nhận ( $T_4$ )

#### 3.2. Cách làm:

- Độ lệch thời gian giữa máy trạm và máy chủ thời gian;

$$\theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

- Thời gian các bản tin lưu chuyển trên mạng
  - $(T_4 - T_1) - (T_3 - T_2)$
- Thời gian máy khách ngay sau khi hoàn thành đồng bộ:
  - $T = T_4 + \theta$

#### 3.3. Mẹo tính toán:

- Các thời gian sẽ giống nhau đều ở format: yyyy-mm-dd hh:mi:ss.ms
- Chuyển hết thời gian về ms → tính theo ms → chuyển lại về đúng format
- So sánh xem nó giống nhau tới phần nào yyyy, mm, dd, hh, ... thì tính toán chỉ cần quan tâm đến phần còn lại.
- Làm tròn:  $\leq 0.5$ ,  $-0.5 \rightarrow$  làm tròn xuống

#### 3.4. Mẹo tính máy tính

- Chuyển thời gian → ms và lưu vào các biến trong máy tính.
- Lấy các thời gian đã lưu trong các biến của máy tính rồi áp dụng công thức tính

- Chuyển lại thời gian cho đúng định dạng.

## 4. Giải thuật Berkeley

### 4.1. Đề bài

- Có N tiến trình, và thời gian của mỗi tiến trình, tiến trình  $P_i$  làm điều phối.

### 4.2. Cách làm

- Lặp lần lượt tất cả các tiến trình  $P_j$ 
  - o Độ lệch = Thời gian của  $P_j$  - Thời gian của tiến trình  $P_i$
- Tính trung bình cộng các độ lệch vừa tính được
- Giá trị điều chỉnh của mỗi tiến trình = Giá trị trung bình cộng – Độ lệch
- Thời gian sau khi đồng bộ = Thời gian của một tiến trình bất kỳ + giá trị điều chỉnh của tiến trình đó.
- *Chú ý:*
  - o *Sau khi đồng bộ thì các tiến trình có thời gian bằng nhau*
  - o *Kết quả tính trung bình làm tròn như sau*
    - $\leq 0.5, -0.5 \rightarrow$  Làm tròn xuống

### 4.3. Mẹo tính máy tính

- o Chuyển thời gian  $\rightarrow$  ms và lưu vào trong máy tính
- o Áp dụng công thức tính trung bình.
- o Lưu biến trung bình vào máy tính
- o Tính giá trị điều chỉnh và thời gian
- o Nhớ chuyển lại thời gian về đúng format

## 5. Giải thuật trung bình

### 5.1. Đề bài

- Có N tiến trình và thời gian của mỗi tiến trình.

### 5.2. Cách làm

- $T$  = tổng thời gian của các tiến trình.
- Với mỗi tiến trình  $P_i$  thời gian sau khi đồng bộ tính như sau:
  - o Loại bỏ thời gian của  $P_i$  (loại bỏ tạm thời, cuối mỗi lần lặp tập thời gian của các tiến trình lại như cũ).
  - o Tìm thời gian max, min
  - o Thời gian sau khi đồng bộ =  $\frac{T - \max - \min - \text{Time } P_i}{N - 3}$

### 5.3. Máy tính máy tính

- Các thời gian sẽ giống nhau đều ở format: yyyy-mm-dd hh:mi:ss.ms
- So sánh xem nó giống nhau tới phần nào yyyy, mm, dd, hh, ... thì tính toán chỉ cần quan tâm đến phần còn lại.
- Lưu sẵn giá trị 60 x 60 x 1000 hoặc 60 x 1000 tùy vào đề bài.
- Tính tổng thời gian  $T$  như sau:
  - o hh = (Giống phần hh thẳng xuống + tổng với nhau) x 60 x 60 x 1000
  - o mi = (Giống phần mi thẳng xuống + tổng với nhau) x 60 x 1000
  - o ss.ms = Giống phần ss.ms thẳng xuống (bỏ .) = tổng với nhau
  - o  $T = \text{hh} + \text{mi} + \text{ss.ms}$
- Lúc tìm max min cũng giống xuống rồi so sánh từng phần.
  - o Lưu sẵn hai biến max, min vào máy tính
  - o Lưu sẵn cả  $\text{max2} < \text{max}$ ,  $\text{min2} < \text{min}$



## **6. Giao thức RBS**

### **6.1. Đề bài**

- Tiến trình P thực hiện M lần gửi tín hiệu đồng bộ thời gian vật lý đến tiến trình Q
- Thời gian mỗi lần tiến trình P gửi tín hiệu thứ i.
- Thời gian mỗi lần tiến trình Q nhận tín hiệu thứ i

### **6.2. Cách làm**

- Độ lệch của mỗi tiến trình = thời gian của tiến trình P – thời gian của tiến trình Q

### **6.3. Cách bấm máy tính**

- Tạo một biểu thức tổng quát:
  - $((A \times 60 \times 60 \times 1000 + B \times 60 \times 1000 + C) - (D \times 60 \times 60 \times 1000 + E \times 60 \times 1000 + F)) \times 10000$  (Ticks)
  - Tùy theo giống nhau đến phần nào thì có thể bỏ A, D hoặc B, C , ... đi

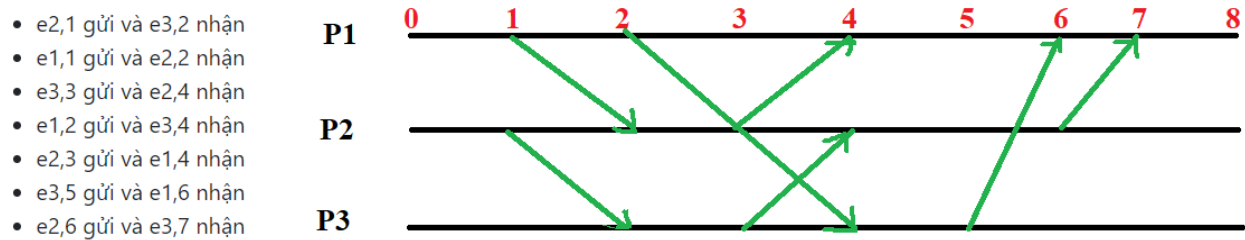
## 7. Nhãn thời gian vector

### 7.1. Đề bài

- Có N tiến trình, M sự kiện
- Thông điệp giữa các tiến trình
  - o  $e(i', j')$  gửi và  $e(i, j)$  nhận
  - o ...
- Có các nhãn vector khởi tạo tại sự kiện số 0

### 7.2. Cách làm

- Vẽ các sự kiện ra



- Xét từng ô  $(i, j)$ 
  - o  $\hat{O}(i, j) = \hat{O}(i, j-1)$  tăng phần tử thứ  $i$  trong vector lên 1 đơn vị
  - o Nếu có “ $e(i', j')$  gửi và  $e(i, j)$  nhận” thì
    - So sánh từng phần tử  $i, j$  rồi lấy  $\max \rightarrow \hat{O}(i, j)$  bằng bộ vector mới.

## 8. Nhãn thời gian lamport

### 8.1. Đề bài

- Có N tiến trình, và M sự kiện
- Thông điệp giữa các tiến trình
  - o  $e(i', j')$  gửi  $e(i, j)$  với  $(1 \leq i, i' \leq N, 1 \leq j, j' \leq M)$
  - o ...
- Bộ đếm thời gian của mỗi tiến trình:
  - o Là thời gian xảy ra sự kiện tại 1 tiến trình cụ thể nào đó

Thông điệp giữa các tiến trình	
<ul style="list-style-type: none"><li>• e1,1 gửi và e2,2 nhận</li><li>• e2,3 gửi và e3,4 nhận</li><li>• e3,6 gửi và e2,7 nhận</li><li>• e2,8 gửi và e1,9 nhận</li></ul>	

Bộ đếm thời gian của mỗi tiến trình											
TT	eX,0	eX,1	eX,2	eX,3	eX,4	eX,5	eX,6	eX,7	eX,8	eX,9	eX,10
P3	0	10	20	30	40	50	60	70	80	90	100
P2	0	8	16	24	32	40	48	56	64	72	80
P1	0	6	12	18	24	30	36	42	48	54	60

### 8.2. Cách làm

- Sắp xếp các thông điệp theo thứ tự sự kiện (Thông điệp của sự kiện nhỏ hơn sẽ nằm trước, nếu cùng thứ tự sự kiện thì sắp xếp theo thứ tự tiến trình).
- Xét từng sự kiện  $j$  ( $1 \leq j \leq M$ )
  - o Xét từng ô của  $P_i$  trong cột  $j$  ( $1 \leq i \leq N$ )
    - Nếu không có  $e(i', j')$  gửi đến  $e(i, j)$  thì ô(i, j) giữ nguyên kết quả
    - Nếu có:
      - Ô (i, j) mới =  $\max(\text{Ô}(i, j) \text{ cũ}, \text{ô}(i', j') + 1)$
      - Offset = ô (i, j) mới – ô (i, j) cũ
      - Các ô từ ô (i, j + 1) đến ô (i, M) đều cộng thêm offset
- Các ô còn lại chưa điền thì giữ nguyên

### 8.3. Mẹo làm nhanh

- Điền y nguyên bảng ban đầu vào bảng kết quả
- *Bài này có ít event thì làm được như này. Chứ có nhiều event thì phải vẽ ra giống bài nhãn thời gian vector*
- Sắp xếp các sự kiện.

- Xét từng  $e(i, j)$  nhận.
  - Các ô có cột  $\leq$  cột  $j$  thì giữ nguyên.
  - Điền ô  $(i, j)$ :
    - Ô  $(i, j)$  mới =  $\max(\text{Ô}(i, j) \text{ cũ}, \text{ô}(i', j') + 1)$
    - Offset = ô  $(i, j)$  mới – ô  $(i, j)$  cũ
    - Các ô từ ô  $(i, j + 1)$  đến ô  $(i, M)$  đều cộng thêm offset.
- Các ô còn lại chưa điền thì giữ nguyên.

## 9. Bầu chọn không dây

### 9.1. Đề bài

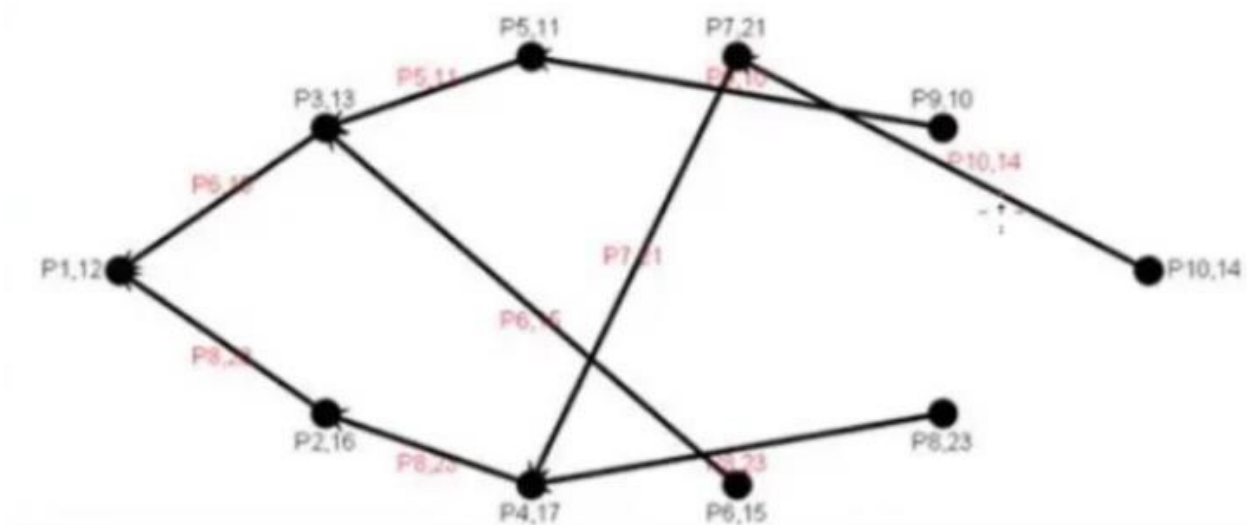
- Cho N tiến trình. Mỗi tiến trình thể hiện bằng cặp tên và giá trị tham gia bầu chọn của tiến trình đó kèm theo mỗi quan hệ cha con với các tiến trình khác như sau (Ví dụ minh họa):

Tiến trình (gửi)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Giá trị ứng cử	11	15	14	12	21	25	18	22	29	32	36	24	38	41	31	33	45	48	37	40
Tiến trình cha (nhận)		P1	P1	P2	P2	P3	P3	P3	P4	P4	P5	P5	P5	P6	P6	P7	P7	P7	P8	P10

- Điền các thông điệp bầu chọn theo định dạng cặp tên tiến trình và giá trị bầu chọn phân cách bằng dấu phẩy (Ví dụ P5,10).

### 9.2. Cách làm

- Vẽ hình ra, kiểu dạng như này (đây chỉ là ví dụ minh họa, không phải cho đề bài bên trên)



- Xét từng mũi tên trong hình (Đi từ mũi tên không có mũi tên nào cắm vào rồi tiến vào trong):
  - o Mỗi mũi tên sẽ tương ứng với “P<sub>i</sub> gửi P<sub>j</sub>”.
  - o Giá trị của mũi tên = Max((P<sub>i</sub>, value), các (P<sub>k</sub>, value) của các mũi tên cắm vào P<sub>i</sub>) (So sánh theo value).

- Giá trị bầu chọn: Là cặp  $(P_i, \text{value})$  lớn nhất cắm vào cái cuối cùng

### 9.3. Cách làm nhanh

- Nếu đề bài cho tiến trình cha  $P_i$  và tiến trình gửi  $P_j$  mà  $i$  luôn nhỏ hơn  $j$  thì sắp xếp bảng đề bài theo thứ tự tăng dần của tiến trình cha.

Tiến trình (gửi)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Giá trị ứng cử	11	15	14	12	21	25	18	22	29	32	36	24	38	41	31	33	45	48	37	40
Tiến trình cha (nhận)		P1	P1	P2	P2	P3	P3	P3	P4	P4	P5	P5	P5	P6	P6	P7	P7	P7	P8	P10

- Ví dụ ở đây ra thấy P10 ở tiến trình cha là lớn nhất  $\rightarrow$  Ta giống tương ứng lên tiến trình gửi.
- Các tiến trình từ P10 đổ lên thì gửi y nguyên giá trị.
  - o P11 gửi P5: P11,36
- Các tiến trình  $\leq$  P10 đổ xuống thì xét  $P_i$  gửi  $P_j$ 
  - o Lấy giá trị max của  $((P_i, \text{value}),$  và các giá trị của  $(P_k, \text{value})$  gửi  $P_i$ )
- Tiến trình bầu chọn là giá trị max của các giá trị gửi cho P1
- **Cách làm nhanh đối với dạng điền bảng ma trận  $N \times N$** 
  - o Các tiến trình từ P10 đổ lên thì gửi y nguyên giá trị.
  - o Các tiến trình  $\leq$  P10 đổ xuống thì xét  $P_i$  gửi  $P_j$ 
    - Lấy giá trị max(cột  $P_i$ , và giá trị của  $P_i$  gửi  $P_j$  trong bảng đề bài)

## 10. Đồng thuận phân tán (Byzantine)

### 10.1. Đề bài

- Cho các tiến trình lỗi và không lỗi

Tiến trình	Tiến trình 1	Tiến trình 2	Tiến trình 3	Tiến trình 4
Giá trị	180	281	382	483
Trạng thái	Không lỗi	Lỗi	Không lỗi	Không lỗi

### 10.2. Cách làm

- Kết quả lần thứ nhất:
  - o Lập từng tiến trình  $P_i$ :
    - $P_i$  gửi cho các tiến trình  $P_j$  khác xem phản hồi lại không.
      - $P_j$  không lỗi  $\Rightarrow$  Có phản hồi lại giá trị của nó
      - $P_j$  lỗi
        - o  $P_i == P_j \Rightarrow$  Phản hồi giá trị của nó
        - o  $P_i != P_j \Rightarrow$  Phản hồi là Nil
    - $\rightarrow$  Thu được 1 vector
- Kết quả lần thứ hai:
  - o Mỗi tiến trình sẽ gửi vector (kết quả bước 1) cho các tiến trình khác.
    - Nếu phản hồi lại thì lấy vector của tiến trình khác đó
    - Không thì sẽ thành Nil hết
- Kết quả cuối cùng:
  - o Ở kết quả lần 2 ta giống dọc xuống. Sẽ thu được kết quả cuối.
    - Giá trị nào ở cột đó chiếm ưu thế thì lấy
    - Nếu giá trị chiếm ưu thế là Nil thì ta điền Unknown

Các kết quả	Tiến trình 1	Tiến trình 2	Tiến trình 3	Tiến trình 4
Kết quả lần thứ nhất	180,Nil,382,483	180,281,382,483	180,Nil,382,483	180,Nil,382,483
Kết quả lần thứ hai	Nil,Nil,Nil,Nil	180,Nil,382,483	180,Nil,382,483	180,Nil,382,483
	180,Nil,382,483	180,Nil,382,483	Nil,Nil,Nil,Nil	Nil,Nil,Nil,Nil
	180,Nil,382,483	180,Nil,382,483	180,Nil,382,483	180,Nil,382,483
Kết quả cuối cùng	180,Unknown,382,483	180,Unknown,382,483	180,Unknown,382,483	180,Unknown,382,483

## 11. Giải thuật không tập trung (Probabilities)

### 11.1. Đề bài

- Mỗi tài nguyên sẽ được nhân bản N lần
- Tiến trình chỉ được cấp quyền truy cập khi nhận được số phiếu  $\geq N/2$
- Số lượng tiến trình m
- Số tiến trình lỗi k
- Xác suất bị lỗi của mỗi tiến trình điều phối: p

### 11.2. Cách làm

- Xác suất k trong m tiến trình điều phối bị lỗi là
  - $P(k) = C_m^k p^k (1 - p)^{m-k}$



## 12. Giải thuật Lan truyền ngẫu nhiên (RandomGossips)

### 12.1. Đề bài

- Có N nút

### 12.2. Cách làm

- $P_i$  là xác suất nút P chưa nhận được thông điệp sau vòng thứ i:
- $\Rightarrow P_{i+1}$  là xác suất nút P chưa nhận được thông điệp sau vòng thứ i+1
  - Phương pháp đẩy:  $P_{i+1} = P_i \left(1 - \frac{1}{N-1}\right)^{N \cdot (1 - P_i)}$
  - Phương pháp kéo:  $P_{i+1} = P_i^2$
  - Phương pháp đẩy và kéo:  $P_{i+1} = P_i^2 \cdot P_i \left(1 - \frac{1}{N-1}\right)^{N \cdot (1 - P_i)}$