

# Áp dụng thuật toán Tarjan tìm các BPLT của đồ thị có hướng (ví dụ) => làm 1 bài

**Đồ thị gốc**

Help Clear shift Delete Edit Undo Red Black

Áp dụng thuật toán Tarjan bằng cách duyệt đệ quy theo chiều sâu bắt đầu từ đỉnh 1

Lùi lại 1 bước Số bước: 52

**SCC(1)** -- --

1. Đánh số cho đỉnh 1 và đưa nó vào ngăn xếp S.  
Gán  $num[1] = min\_num[1] = k = 1$ ;  $k++$ ;  
Đưa 1 vào S. Kết quả: S = 1  
Thực hiện đánh số và đưa vào Stack ☒ 1

2. Với các đỉnh kề v của 1: 2,3 Xét ☒ 2

2a. v = '2' chưa duyệt Duyệt nó ☒ 3

**SCC(2)** -- --

1. Đánh số cho đỉnh 2 và đưa nó vào ngăn xếp S.  
Gán  $num[2] = min\_num[2] = k = 2$ ;  $k++$ ;  
Đưa 2 vào S. Kết quả: S = 1,2  
Thực hiện đánh số và đưa vào Stack ☒ 4

2. Với các đỉnh kề v của 2: 1,8 Xét ☒ 5

2b. v = '1' duyệt rồi nhưng vẫn còn trên stack Cập nhật  $min\_num[u]$  ☒ 6  
Cập nhật  $min\_num[2] = \min(min\_num[2], num[1]) = 1$  Cập nhật ☒ 7

2a. v = '8' chưa duyệt Duyệt nó ☒ 8

**SCC(8)** -- --

1. Đánh số cho đỉnh 8 và đưa nó vào ngăn xếp S.  
Gán  $num[8] = min\_num[8] = k = 3$ ;  $k++$ ;  
Đưa 8 vào S. Kết quả: S = 1,2,8  
Thực hiện đánh số và đưa vào Stack ☒ 9

2. Với các đỉnh kề v của 8: 7 Xét ☒ 10

2a. v = '7' chưa duyệt Duyệt nó ☒ 11

**SCC(7)** -- --

1. Đánh số cho đỉnh 7 và đưa nó vào ngăn xếp S.  
Gán  $num[7] = min\_num[7] = k = 4$ ;  $k++$ ;  
Đưa 7 vào S. Kết quả: S = 1,2,8,7  
Thực hiện đánh số và đưa vào Stack ☒ 12

2. Với các đỉnh kề v của 7: 6 Xét ☒ 13

2a. v = '6' chưa duyệt Duyệt nó ☒ 14

**SCC(6)** -- --

1. Đánh số cho đỉnh 6 và đưa nó vào ngăn xếp S.  
Gán  $num[6] = min\_num[6] = k = 5$ ;  $k++$ ;  
Đưa 6 vào S. Kết quả: S = 1,2,8,7,6  
Thực hiện đánh số và đưa vào Stack ☒ 15

2. Với các đỉnh kề v của 6: 7 Xét ☒ 16

2b. v = '7' duyệt rồi nhưng vẫn còn trên stack Cập nhật  $min\_num[u]$  ☒ 17  
Cập nhật  $min\_num[6] = \min(min\_num[6], num[7]) = 4$  Cập nhật ☒ 18

3. Kiểm tra num và  $min\_num$  của 6  
 $num[6] == min\_num[6]$   $num[6] != min\_num[6]$  ☒ 19

Cập nhật  $min\_num[7] = \min(min\_num[7], min\_num[6]) = 4$  Cập nhật ☒ 20

3. Kiểm tra num và  $min\_num$  của 7  
 $num[7] == min\_num[7]$   $num[7] != min\_num[7]$  ☒ 21  
Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được 7  
Các đỉnh được lấy ra: 6,7  
Nội dung còn lại của ngăn xếp: 1,2,8  
Cập nhật ☒ 22

Cập nhật  $min\_num[8] = \min(min\_num[8], min\_num[7]) = 3$  Cập nhật ☒ 23

3. Kiểm tra num và  $min\_num$  của 8  
 $num[8] == min\_num[8]$   $num[8] != min\_num[8]$  ☒ 24  
Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được 8  
Các đỉnh được lấy ra: 8  
Nội dung còn lại của ngăn xếp: 1,2  
Cập nhật ☒ 25

Cập nhật  $min\_num[2] = \min(min\_num[2], min\_num[8]) = 1$  Cập nhật ☒ 26

3. Kiểm tra num và  $min\_num$  của 2  
 $num[2] == min\_num[2]$   $num[2] != min\_num[2]$  ☒ 27

Cập nhật min\_num[1] = min(min\_num[1], min\_num[2]) = 1  ☒ 28

2a. v = '3' chưa duyệt  ☒ 29

SCC(3)

1. Đánh số cho đỉnh 3 và đưa nó vào ngăn xếp S.  
 Gán num[3] = min\_num[3] = k = 6  : k++;  
 Đưa 3 vào S. Kết quả S = 1,2,3  
 ☒ 30

2. Với các đỉnh kề v của 3: 4,5  ☒ 31

2a. v = '4' chưa duyệt  ☒ 32

SCC(4)

1. Đánh số cho đỉnh 4 và đưa nó vào ngăn xếp S.  
 Gán num[4] = min\_num[4] = k = 7  : k++;  
 Đưa 4 vào S. Kết quả S = 1,2,3,4  
 ☒ 33

2. Với các đỉnh kề v của 4: 2,7,8  ☒ 34

2b. v = '2' duyệt rồi nhưng vẫn còn trên stack  ☒ 35  
 Cập nhật min\_num[4] = min(min\_num[4], num[2]) = 2  ☒ 36

2c. v = '7' duyệt rồi và không còn trên stack  ☒ 37

2c. v = '8' duyệt rồi và không còn trên stack  ☒ 38

3. Kiểm tra num và min\_num của 4  
 num[4] == min\_num[4]  ☒ 39

Cập nhật min\_num[3] = min(min\_num[3], min\_num[4]) = 2  ☒ 40

2a. v = '5' chưa duyệt  ☒ 41

SCC(5)

1. Đánh số cho đỉnh 5 và đưa nó vào ngăn xếp S.  
 Gán num[5] = min\_num[5] = k = 8  : k++;  
 Đưa 5 vào S. Kết quả S = 1,2,3,4,5  
 ☒ 42

2. Với các đỉnh kề v của 5: 3,7  ☒ 43

2b. v = '3' duyệt rồi nhưng vẫn còn trên stack  ☒ 44  
 Cập nhật min\_num[5] = min(min\_num[5], num[3]) = 6  ☒ 45

2c. v = '7' duyệt rồi và không còn trên stack  ☒ 46

3. Kiểm tra num và min\_num của 5  
 num[5] == min\_num[5]  ☒ 47

Cập nhật min\_num[3] = min(min\_num[3], min\_num[5]) = 2  ☒ 48

3. Kiểm tra num và min\_num của 3  
 num[3] == min\_num[3]  ☒ 49

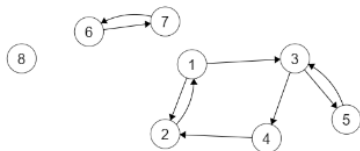
Cập nhật min\_num[1] = min(min\_num[1], min\_num[3]) = 1  ☒ 50

3. Kiểm tra num và min\_num của 1  
 num[1] == min\_num[1]  ☒ 51

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được 1  
 Các đỉnh được lấy ra: 5,4,3,2,1  
 Nội dung còn lại của ngăn xếp:   
 ☒ 52

#### Về các thành phần liên thông

Help Clear shift Delete Edit Undo Red Black



Test	Got	
<input checked="" type="checkbox"/> Test tự động	1. Kiểm tra DFS - Bước 1. SCC(1): 1. Đánh số và đưa u vào ngăn xếp, với u = 1 - Bước 2. SCC(1): 2. Xử lý các đỉnh kề của 1 - Bước 3. SCC(1): 2a. Duyệt 2 - Bước 4. SCC(2): 1. Đánh số và đưa u vào ngăn xếp, với u = 2 - Bước 5. SCC(2): 2. Xử lý các đỉnh kề của 2 - Bước 6. SCC(2): 2b. Cập nhật min_id của 1 - Bước 7. SCC(2): 2b. Cập nhật min_id của 1 - Bước 8. SCC(2): 2a. Duyệt 8 - Bước 9. SCC(8): 1. Đánh số và đưa u vào ngăn xếp, với u = 8 - Bước 10. SCC(8): 2. Xử lý các đỉnh kề của 8 - Bước 11. SCC(8): 2a. Duyệt 7 - Bước 12. SCC(7): 1. Đánh số và đưa u vào ngăn xếp, với u = 7 - Bước 13. SCC(7): 2. Xử lý các đỉnh kề của 7 - Bước 14. SCC(7): 2a. Duyệt 6 - Bước 15. SCC(6): 1. Đánh số và đưa u vào ngăn xếp, với u = 6	<input checked="" type="checkbox"/>

# Áp dụng thuật toán Tarjan tìm các BPLT của đồ thị có hướng (ngẫu nhiên) => làm 4 bài ngẫu nhiên

Đồ thị gốc

Help

Clear

shift

Delete

Edit

Undo

Red

Black

Áp dụng thuật toán Tarjan bằng cách duyệt đệ quy theo chiều sâu bắt đầu từ đỉnh C

❌ Lùi lại 1 bước

Số bước: 45

SCC(C)

1. Đánh số cho đỉnh C và đưa nó vào ngăn xếp S.  
Gán  $num[C] = min\_num[C] = k = 1$ ;  $k++$ ;  
Đưa C vào S. Kết quả: S = C  

Thực hiện đánh số và đưa vào Stack

 1

2. Với các đỉnh kề v của C: A, G 

Xét

 2

2a. v = 'A' chưa duyệt 

Duyệt nó

 3

SCC(A)

1. Đánh số cho đỉnh A và đưa nó vào ngăn xếp S.  
Gán  $num[A] = min\_num[A] = k = 2$ ;  $k++$ ;  
Đưa A vào S. Kết quả: S = C, A  

Thực hiện đánh số và đưa vào Stack

 4

2. Với các đỉnh kề v của A: B 

Xét

 5

2a. v = 'B' chưa duyệt 

Duyệt nó

 6

SCC(B)

1. Đánh số cho đỉnh B và đưa nó vào ngăn xếp S.  
Gán  $num[B] = min\_num[B] = k = 3$ ;  $k++$ ;  
Đưa B vào S. Kết quả: S = C, A, B  

Thực hiện đánh số và đưa vào Stack

 7

2. Với các đỉnh kề v của B: A, F 

Xét

 8

2b. v = 'A' duyệt rồi nhưng vẫn còn trên stack 

Cập nhật min\_num[u]

 9  

Cập nhật  $min\_num[B] = \min(min\_num[B], num[A]) = 2$ 

Cập nhật

 10

2a. v = 'F' chưa duyệt 

Duyệt nó

 11

SCC(F)

1. Đánh số cho đỉnh F và đưa nó vào ngăn xếp S.  
Gán  $num[F] = min\_num[F] = k = 4$ ;  $k++$ ;  
Đưa F vào S. Kết quả: S = C, A, B, F  

Thực hiện đánh số và đưa vào Stack

 12

2. Với các đỉnh kề v của F: C, D 

Xét

 13

2b. v = 'C' duyệt rồi nhưng vẫn còn trên stack 

Cập nhật min\_num[u]

 14  

Cập nhật  $min\_num[F] = \min(min\_num[F], num[C]) = 1$ 

Cập nhật

 15

2a. v = 'D' chưa duyệt 

Duyệt nó

 16

SCC(D)

1. Đánh số cho đỉnh D và đưa nó vào ngăn xếp S.  
Gán  $num[D] = min\_num[D] = k = 5$ ;  $k++$ ;  
Đưa D vào S. Kết quả: S = C, A, B, F, D  

Thực hiện đánh số và đưa vào Stack

 17

2. Với các đỉnh kề v của D: A, E 

Xét

 18

2b. v = 'A' duyệt rồi nhưng vẫn còn trên stack 

Cập nhật min\_num[u]

 19  

Cập nhật  $min\_num[D] = \min(min\_num[D], num[A]) = 2$ 

Cập nhật

 20

2a. v = 'E' chưa duyệt 

Duyệt nó

 21

SCC(E)

1. Đánh số cho đỉnh E và đưa nó vào ngăn xếp S.  
Gán  $num[E] = min\_num[E] = k = 6$ ;  $k++$ ;  
Đưa E vào S. Kết quả: S = C, A, B, F, D, E  

Thực hiện đánh số và đưa vào Stack

 22

2. Với các đỉnh kề v của E: 

Xét

 23

3. Kiểm tra num và min\_num của E  

$num[E] == min\_num[E]$ 

num[E] != min\_num[E]

 24

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được E  
Các đỉnh được lấy ra: E  
Nội dung còn lại của ngăn xếp: C, A, B, F, D  

Cập nhật

 25

Cập nhật  $min\_num[D] = \min(min\_num[D], min\_num[E]) = 2$ 

Cập nhật

 26

3. Kiểm tra num và min\_num của D  

$num[D] == min\_num[D]$ 

num[D] != min\_num[D]

 27

Cập nhật  $min\_num[F] = \min(min\_num[F], min\_num[D]) = 1$ 

Cập nhật

 28

3. Kiểm tra num và min\_num của F

num[F] == min\_num[F]

num[F] != min\_num[F]

X 29

Cập nhật min\_num[B] = min(min\_num[B], min\_num[F]) = 1

Cập nhật 30

3. Kiểm tra num và min\_num của B

num[B] == min\_num[B]

num[B] != min\_num[B]

X 31

Cập nhật min\_num[A] = min(min\_num[A], min\_num[B]) = 1

Cập nhật 32

3. Kiểm tra num và min\_num của A

num[A] == min\_num[A]

num[A] != min\_num[A]

X 33

Cập nhật min\_num[C] = min(min\_num[C], min\_num[A]) = 1

Cập nhật 34

2a. v = 'G' chưa duyệt

Duyệt nó 35

SCC(G) -- --

1. Đánh số cho đỉnh G và đưa nó vào ngăn xếp S.

Gán num[G] = min\_num[G] = k = 7; k++;

Đưa G vào S. Kết quả S = C,A,B,F,D,G

Thực hiện đánh số và đưa vào Stack 36

2. Với các đỉnh kề v của G: D,F

Xét 37

2b. v = 'D' duyệt rồi nhưng vẫn còn trên stack

Cập nhật min\_num[u] 38

Cập nhật min\_num[G] = min(min\_num[G], num[D]) = 5

Cập nhật 39

2b. v = 'F' duyệt rồi nhưng vẫn còn trên stack

Cập nhật min\_num[u] 40

Cập nhật min\_num[G] = min(min\_num[G], num[F]) = 4

Cập nhật 41

3. Kiểm tra num và min\_num của G

num[G] == min\_num[G]

num[G] != min\_num[G]

X 42

Cập nhật min\_num[C] = min(min\_num[C], min\_num[G]) = 1

Cập nhật 43

3. Kiểm tra num và min\_num của C

num[C] == min\_num[C]

num[C] != min\_num[C]

44

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được C

Các đỉnh được lấy ra: G,D,F,B,A,C

Nội dung còn lại của ngăn xếp:

Cập nhật 45



Test	Got	
<div> <div>Test tự động</div> <div> <div>1. Kiểm tra DFS</div> <div> <div>Bước 1. SCC(C): 1. Đánh số và đưa u vào ngăn xếp, với u = C</div> <div>Bước 2. SCC(C): 2. Xử lý các đỉnh kề của C</div> <div>Bước 3. SCC(C): 2a. Duyệt A</div> <div>Bước 4. SCC(A): 1. Đánh số và đưa u vào ngăn xếp, với u = A</div> <div>Bước 5. SCC(A): 2. Xử lý các đỉnh kề của A</div> <div>Bước 6. SCC(A): 2a. Duyệt B</div> <div>Bước 7. SCC(B): 1. Đánh số và đưa u vào ngăn xếp, với u = B</div> <div>Bước 8. SCC(B): 2. Xử lý các đỉnh kề của B</div> <div>Bước 9. SCC(B): 2b. Cập nhật min_id của A</div> <div>Bước 10. SCC(B): 2b. Cập nhật min_id của A</div> <div>Bước 11. SCC(B): 2a. Duyệt F</div> <div>Bước 12. SCC(F): 1. Đánh số và đưa u vào ngăn xếp, với u = F</div> <div>Bước 13. SCC(F): 2. Xử lý các đỉnh kề của F</div> <div>Bước 14. SCC(F): 2b. Cập nhật min_id của C</div> <div>Bước 15. SCC(F): 2b. Cập nhật min_id của C</div> <div>Bước 16. SCC(F): 2a. Duyệt D</div> <div>Bước 17. SCC(D): 1. Đánh số và đưa u vào ngăn xếp, với u = D</div> <div>Bước 18. SCC(D): 2. Xử lý các đỉnh kề của D</div> <div>Bước 19. SCC(D): 2b. Cập nhật min_id của A</div> <div>Bước 20. SCC(D): 2b. Cập nhật min_id của A</div> <div>Bước 21. SCC(D): 2a. Duyệt E</div> <div>Bước 22. SCC(E): 1. Đánh số và đưa u vào ngăn xếp, với u = E</div> <div>Bước 23. SCC(E): 2. Xử lý các đỉnh kề của E</div> </div> </div> </div>		

Dữ thế gốc

Help

Clear

shift

Delete

Edit

Undo

Red

Black

Áp dụng thuật toán Tarjan bằng cách duyệt đệ quy theo chiều sâu bắt đầu từ đỉnh A

[n Lùi lại 1 bước](#) Số bước: 33

SCC(A)

1. Đánh số cho đỉnh A và đưa nó vào ngăn xếp S.  
Gán  $num[A] = min\_num[A] = k = 1$ ;  $k++$ ;  
Đưa A vào S. Kết quả: S = A  
[Thực hiện đánh số và đưa vào Stack](#) ☒ 1

2. Với các đỉnh kề v của A: B, D, E  Xét ☒ 2

2a. v = 'B' chưa duyệt: [Duyệt nó](#) ☒ 3

SCC(B)

1. Đánh số cho đỉnh B và đưa nó vào ngăn xếp S.  
Gán  $num[B] = min\_num[B] = k = 2$ ;  $k++$ ;  
Đưa B vào S. Kết quả: S = A, B  
[Thực hiện đánh số và đưa vào Stack](#) ☒ 4

2. Với các đỉnh kề v của B: A, C  Xét ☒ 5

2b. v = 'A' duyệt rồi nhưng vẫn còn trên stack: [Cập nhật  \$min\\_num\[u\]\$](#)  ☒ 6  
[Cập nhật  \$min\\_num\[B\] = \min\(min\\_num\[B\], num\[A\]\) = 1\$](#)  [Cập nhật](#) ☒ 7

2a. v = 'C' chưa duyệt: [Duyệt nó](#) ☒ 8

SCC(C)

1. Đánh số cho đỉnh C và đưa nó vào ngăn xếp S.  
Gán  $num[C] = min\_num[C] = k = 3$ ;  $k++$ ;  
Đưa C vào S. Kết quả: S = A, B, C  
[Thực hiện đánh số và đưa vào Stack](#) ☒ 9

2. Với các đỉnh kề v của C: D, E  Xét ☒ 10

2a. v = 'D' chưa duyệt: [Duyệt nó](#) ☒ 11

SCC(D)

1. Đánh số cho đỉnh D và đưa nó vào ngăn xếp S.  
Gán  $num[D] = min\_num[D] = k = 4$ ;  $k++$ ;  
Đưa D vào S. Kết quả: S = A, B, C, D  
[Thực hiện đánh số và đưa vào Stack](#) ☒ 12

2. Với các đỉnh kề v của D:  Xét ☒ 13

3. Kiểm tra  $num$  và  $min\_num$  của D  
 $num[D] == min\_num[D]$  ☒  $num[D] != min\_num[D]$  ☒ 14  
Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được D  
Các đỉnh được lấy ra: D  
Nội dung còn lại của ngăn xếp: A, B, C  
[Cập nhật](#) ☒ 15

[Cập nhật  \$min\\_num\[C\] = \min\(min\\_num\[C\], min\\_num\[D\]\) = 3\$](#)  [Cập nhật](#) ☒ 16

2a. v = 'E' chưa duyệt: [Duyệt nó](#) ☒ 17

SCC(E)

1. Đánh số cho đỉnh E và đưa nó vào ngăn xếp S.  
Gán  $num[E] = min\_num[E] = k = 5$ ;  $k++$ ;  
Đưa E vào S. Kết quả: S = A, B, C, E  
[Thực hiện đánh số và đưa vào Stack](#) ☒ 18

2. Với các đỉnh kề v của E: B, D  Xét ☒ 19

2b. v = 'B' duyệt rồi nhưng vẫn còn trên stack: [Cập nhật  \$min\\_num\[u\]\$](#)  ☒ 20  
[Cập nhật  \$min\\_num\[E\] = \min\(min\\_num\[E\], num\[B\]\) = 2\$](#)  [Cập nhật](#) ☒ 21

2c. v = 'D' duyệt rồi và không còn trên stack: [Bỏ qua](#) ☒ 22

3. Kiểm tra  $num$  và  $min\_num$  của E  
 $num[E] == min\_num[E]$  ☒  $num[E] != min\_num[E]$  ☒ 23

[Cập nhật  \$min\\_num\[C\] = \min\(min\\_num\[C\], min\\_num\[E\]\) = 2\$](#)  [Cập nhật](#) ☒ 24

3. Kiểm tra  $num$  và  $min\_num$  của C  
 $num[C] == min\_num[C]$  ☒  $num[C] != min\_num[C]$  ☒ 25

[Cập nhật  \$min\\_num\[B\] = \min\(min\\_num\[B\], min\\_num\[C\]\) = 1\$](#)  [Cập nhật](#) ☒ 26

3. Kiểm tra  $num$  và  $min\_num$  của B  
 $num[B] == min\_num[B]$  ☒  $num[B] != min\_num[B]$  ☒ 27

[Cập nhật  \$min\\_num\[A\] = \min\(min\\_num\[A\], min\\_num\[B\]\) = 1\$](#)  [Cập nhật](#) ☒ 28

2c. v = 'D' duyệt rồi và không còn trên stack

Bỏ qua

29

2b. v = 'E' duyệt rồi nhưng vẫn còn trên stack

Cập nhật min\_num[u]

30

Cập nhật min\_num[A] = min(min\_num[A], num[E]) = 1

Cập nhật

31

3. Kiểm tra num và min\_num của A

num[A] == min\_num[A]

num[A] != min\_num[A]

32

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được A

Các đỉnh được lấy ra: E,C,B,A

Nội dung còn lại của ngăn xếp:

Cập nhật

33

Vẽ các thành phần liên thông

Help

Clear

shift

Delete

Edit

Undo

Red

Black

```
graph TD; A --> B; B --> A; A --> E; E --> C; C --> B; D --- D;
```

Test	Got	
✓	<div>Test tự động</div> <div>1. Kiểm tra DFS</div> <div>- Bước 1. SCC(A): 1. Đánh số và đưa u vào ngăn xếp, với u = A</div> <div>- Bước 2. SCC(A): 2. Xử lý các đỉnh kề của A</div> <div>- Bước 3. SCC(A): 2a. Duyệt B</div> <div>- Bước 4. SCC(B): 1. Đánh số và đưa u vào ngăn xếp, với u = B</div> <div>- Bước 5. SCC(B): 2. Xử lý các đỉnh kề của B</div> <div>- Bước 6. SCC(B): 2b. Cập nhật min_id của A</div> <div>- Bước 7. SCC(B): 2b. Cập nhật min_id của A</div> <div>- Bước 8. SCC(B): 2a. Duyệt C</div> <div>- Bước 9. SCC(C): 1. Đánh số và đưa u vào ngăn xếp, với u = C</div> <div>- Bước 10. SCC(C): 2. Xử lý các đỉnh kề của C</div> <div>- Bước 11. SCC(C): 2a. Duyệt D</div> <div>- Bước 12. SCC(D): 1. Đánh số và đưa u vào ngăn xếp, với u = D</div> <div>- Bước 13. SCC(D): 2. Xử lý các đỉnh kề của D</div> <div>- Bước 14. SCC(D): 3a. id[u] == min_id[u] với u = D</div> <div>- Bước 15. SCC(D): 3a. Tìm được SCC của D</div> <div>- Bước 16. SCC(C): 2a. Cập nhật min_id của D</div> <div>- Bước 17. SCC(C): 2a. Duyệt E</div> <div>- Bước 18. SCC(E): 1. Đánh số và đưa u vào ngăn xếp, với u = E</div> <div>- Bước 19. SCC(E): 2. Xử lý các đỉnh kề của E</div> <div>- Bước 20. SCC(E): 2b. Cập nhật min_id của B</div> <div>- Bước 21. SCC(E): 2b. Cập nhật min_id của B</div> <div>- Bước 22. SCC(E): 2c. Bỏ qua đỉnh D</div> <div>- Bước 23. SCC(E): 2b. id[u] != min_id[u] với u = E</div> <div>- Bước 24. SCC(C): 2a. Cập nhật min_id của E</div>	✓

Đồ thị gốc

Help

Clear

shift

Delete

Edit

Undo

Red

Black

```
graph TD; D --> A; D --> C; A --> B; A --> E; C --> E;
```

Áp dụng thuật toán Tarjan bằng cách duyệt đệ quy theo chiều sâu bắt đầu từ đỉnh D

Lùi lại 1 bước

Số bước: 30

SCC(D) 

--

--

1. Đánh số cho đỉnh D và đưa nó vào ngăn xếp S.

Gán num[D] = min\_num[D] = k = 1; k++;

Đưa D vào S. Kết quả: S = D

Thực hiện đánh số và đưa vào Stack

1

2. Với các đỉnh kề v của D: A,C

Xét

2

2a. v = 'A' chưa duyệt

Duyệt nó

3

SCC(A) 

--

--

1. Đánh số cho đỉnh A và đưa nó vào ngăn xếp S.

Gán num[A] = min\_num[A] = k = 2; k++;

Đưa A vào S. Kết quả: S = D,A

Thực hiện đánh số và đưa vào Stack

4

2. Với các đỉnh kề v của A: B

Xét

5

2a. v = 'B' chưa duyệt

Duyệt nó

6

SCC(B) 

--

--

1. Đánh số cho đỉnh B và đưa nó vào ngăn xếp S.

Gán num[B] = min\_num[B] = k = 3; k++;

Đưa B vào S. Kết quả: S = D,A,B

Thực hiện đánh số và đưa vào Stack

7

2. Với các đỉnh kề v của B:

Xét

8

3. Kiểm tra num và min\_num của B

num[B] == min\_num[B]

num[B] != min\_num[B]

9

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được B

Các đỉnh được lấy ra: B

Nội dung còn lại của ngăn xếp: D,A

Cập nhật

10

Cập nhật min\_num[A] = min(min\_num[A], min\_num[B]) = 2

Cập nhật

11

3. Kiểm tra num và min\_num của A

num[A] == min\_num[A]

num[A] != min\_num[A]

12

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được A

Các đỉnh được lấy ra: A

Nội dung còn lại của ngăn xếp: D

Cập nhật

13

Cập nhật min\_num[D] = min(min\_num[D], min\_num[A]) = 1

Cập nhật

14

2a. v = 'C' chưa duyệt

Duyệt nó

15

SCC(C)

1. Đánh số cho đỉnh C và đưa nó vào ngăn xếp S.

Gán num[C] = min\_num[C] = k = 4

k++

Đưa C vào S. Kết quả: S = D,C

Thực hiện đánh số và đưa vào Stack

16

2. Với các đỉnh kề v của C: A,E

Xét

17

2c. v = 'A' duyệt rồi và không còn trên stack

Bỏ qua

18

2a. v = 'E' chưa duyệt

Duyệt nó

19

SCC(E)

1. Đánh số cho đỉnh E và đưa nó vào ngăn xếp S.

Gán num[E] = min\_num[E] = k = 5

k++

Đưa E vào S. Kết quả: S = D,C,E

Thực hiện đánh số và đưa vào Stack

20

2. Với các đỉnh kề v của E: A

Xét

21

2c. v = 'A' duyệt rồi và không còn trên stack

Bỏ qua

22

3. Kiểm tra num và min\_num của E

num[E] == min\_num[E]

num[E] != min\_num[E]

23

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được E

Các đỉnh được lấy ra: E

Nội dung còn lại của ngăn xếp: D,C

Cập nhật

24

Cập nhật min\_num[C] = min(min\_num[C], min\_num[E]) = 4

Cập nhật

25

3. Kiểm tra num và min\_num của C

num[C] == min\_num[C]

num[C] != min\_num[C]

26

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được C

Các đỉnh được lấy ra: C

Nội dung còn lại của ngăn xếp: D

Cập nhật

27

Cập nhật min\_num[D] = min(min\_num[D], min\_num[C]) = 1

Cập nhật

28

3. Kiểm tra num và min\_num của D

num[D] == min\_num[D]

num[D] != min\_num[D]

29

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được D

Các đỉnh được lấy ra: D

Nội dung còn lại của ngăn xếp:

Cập nhật

30

Về các thành phần liên thông

Help

Clear

Shift

Delete

Edit

Undo

Red

Black

A

D

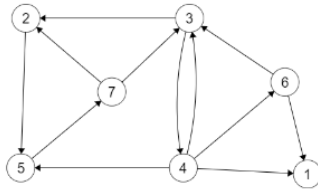
B

C

E

Test	Got	
✓ Test tự động	<div>1. Kiểm tra DFS</div> <div><div>- Bước 1. SCC(D): 1. Đánh số và đưa u vào ngăn xếp, với u = D</div><div>- Bước 2. SCC(D): 2. Xử lý các đỉnh kề của D</div><div>- Bước 3. SCC(D): 2a. Duyệt A</div><div>- Bước 4. SCC(A): 1. Đánh số và đưa u vào ngăn xếp, với u = A</div><div>- Bước 5. SCC(A): 2. Xử lý các đỉnh kề của A</div><div>- Bước 6. SCC(A): 2a. Duyệt B</div><div>- Bước 7. SCC(B): 1. Đánh số và đưa u vào ngăn xếp, với u = B</div><div>- Bước 8. SCC(B): 2. Xử lý các đỉnh kề của B</div><div>- Bước 9. SCC(B): 3a. id[u] == min_id[u] với u = B</div><div>- Bước 10. SCC(B): 3a. Tìm được SCC của B</div><div>- Bước 11. SCC(A): 2a. Cập nhật min_id của B</div><div>- Bước 12. SCC(A): 3a. id[u] == min_id[u] với u = A</div><div>- Bước 13. SCC(A): 3a. Tìm được SCC của A</div><div>- Bước 14. SCC(D): 2a. Cập nhật min_id của A</div><div>- Bước 15. SCC(D): 2a. Duyệt C</div><div>- Bước 16. SCC(C): 1. Đánh số và đưa u vào ngăn xếp, với u = C</div><div>- Bước 17. SCC(C): 2. Xử lý các đỉnh kề của C</div><div>- Bước 18. SCC(C): 2c. Bỏ qua đỉnh A</div><div>- Bước 19. SCC(C): 2a. Duyệt E</div><div>- Bước 20. SCC(E): 1. Đánh số và đưa u vào ngăn xếp, với u = E</div><div>- Bước 21. SCC(E): 2. Xử lý các đỉnh kề của E</div><div>- Bước 22. SCC(E): 2c. Bỏ qua đỉnh A</div><div>- Bước 23. SCC(E): 3a. id[u] == min_id[u] với u = E</div></div> <div>✓</div>	

✓



Áp dụng thuật toán Tarjan bằng cách duyệt đệ quy theo chiều sâu bắt đầu từ đỉnh 3

❌ Lỗi lại 1 bước Số bước: 46

SCC(3)

1. Đánh số cho đỉnh 3 và đưa nó vào ngăn xếp S.

Gán  $num[3] = min\_num[3] = k = 1$ ;  $k++$ ;

Đưa 3 vào S. Kết quả: S = {3}

Thực hiện đánh số và đưa vào Stack ☒ 1

2. Với các đỉnh kề v của 3:

2,4

Xét ☒ 2

2a. v = '2' chưa duyệt ☒ 3

SCC(2)

1. Đánh số cho đỉnh 2 và đưa nó vào ngăn xếp S.

Gán  $num[2] = min\_num[2] = k = 2$ ;  $k++$ ;

Đưa 2 vào S. Kết quả: S = {3,2}

Thực hiện đánh số và đưa vào Stack ☒ 4

2. Với các đỉnh kề v của 2:

5

Xét ☒ 5

2a. v = '5' chưa duyệt ☒ 6

SCC(5)

1. Đánh số cho đỉnh 5 và đưa nó vào ngăn xếp S.

Gán  $num[5] = min\_num[5] = k = 3$ ;  $k++$ ;

Đưa 5 vào S. Kết quả: S = {3,2,5}

Thực hiện đánh số và đưa vào Stack ☒ 7

2. Với các đỉnh kề v của 5:

7

Xét ☒ 8

2a. v = '7' chưa duyệt ☒ 9

SCC(7)

1. Đánh số cho đỉnh 7 và đưa nó vào ngăn xếp S.

Gán  $num[7] = min\_num[7] = k = 4$ ;  $k++$ ;

Đưa 7 vào S. Kết quả: S = {3,2,5,7}

Thực hiện đánh số và đưa vào Stack ☒ 10

2. Với các đỉnh kề v của 7:

2,3

Xét ☒ 11

2b. v = '2' duyệt rồi nhưng vẫn còn trên stack

Cập nhật  $min\_num[2]$  ☒ 12

Cập nhật  $min\_num[7] = \min(min\_num[7], num[2]) = 2$

Cập nhật ☒ 13

2b. v = '3' duyệt rồi nhưng vẫn còn trên stack

Cập nhật  $min\_num[3]$  ☒ 14

Cập nhật  $min\_num[7] = \min(min\_num[7], num[3]) = 1$

Cập nhật ☒ 15

3. Kiểm tra  $num$  và  $min\_num$  của 7

$num[7] == min\_num[7]$  ☒ 16

Cập nhật  $min\_num[5] = \min(min\_num[5], min\_num[7]) = 1$

Cập nhật ☒ 17

3. Kiểm tra  $num$  và  $min\_num$  của 5

$num[5] == min\_num[5]$  ☒ 18

Cập nhật  $min\_num[2] = \min(min\_num[2], min\_num[5]) = 1$

Cập nhật ☒ 19

3. Kiểm tra  $num$  và  $min\_num$  của 2

$num[2] == min\_num[2]$  ☒ 20

Cập nhật  $min\_num[3] = \min(min\_num[3], min\_num[2]) = 1$

Cập nhật ☒ 21

2a. v = '4' chưa duyệt ☒ 22

SCC(4)

1. Đánh số cho đỉnh 4 và đưa nó vào ngăn xếp S.

Gán  $num[4] = min\_num[4] = k = 5$ ;  $k++$ ;

Đưa 4 vào S. Kết quả: S = {3,2,5,7,4}

Thực hiện đánh số và đưa vào Stack ☒ 23

2. Với các đỉnh kề v của 4:

1,3,5,6

Xét ☒ 24

2a. v = '1' chưa duyệt ☒ 25

SCC(1)

1. Đánh số cho đỉnh 1 và đưa nó vào ngăn xếp S.

Gán  $num[1] = min\_num[1] = k = 6$ ;  $k++$ ;

Đưa 1 vào S. Kết quả: S = {3,2,5,7,4,1}

Thực hiện đánh số và đưa vào Stack ☒ 26

2. Với các đỉnh kề v của 1:

Xét ☒ 27

3. Kiểm tra  $num$  và  $min\_num$  của 1

$num[1] == min\_num[1]$  ☒ 28



Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được 1

Các đỉnh được lấy ra: 1

Nội dung còn lại của ngăn xếp: 3,2,5,7,4

Cập nhật 29

Cập nhật min\_num[4] = min(min\_num[4], min\_num[1]) = 5

Cập nhật 30

2b. v = '3' duyệt rồi nhưng vẫn còn trên stack

Cập nhật min\_num[u] 31

Cập nhật min\_num[4] = min(min\_num[4], num[3]) = 1

Cập nhật 32

2b. v = '5' duyệt rồi nhưng vẫn còn trên stack

Cập nhật min\_num[u] 33

Cập nhật min\_num[4] = min(min\_num[4], num[5]) = 1

Cập nhật 34

2a. v = '6' chưa duyệt

Duyệt nó 35

SCC(6) -- --

1. Đỉnh số cho đỉnh 6 và đưa nó vào ngăn xếp S.

Gán num[6] = min\_num[6] = k = 7; k++;

Đưa 6 vào S. Kết quả: S = 3,2,5,7,4,6

Thực hiện đánh số và đưa vào Stack 36

2. Với các đỉnh kề v của 6: 1,3

Xét 37

2c. v = '1' duyệt rồi và không còn trên stack

Bỏ qua 38

2b. v = '3' duyệt rồi nhưng vẫn còn trên stack

Cập nhật min\_num[u] 39

Cập nhật min\_num[6] = min(min\_num[6], num[3]) = 1

Cập nhật 40

3. Kiểm tra num và min\_num của 6

num[6] == min\_num[6] num[6] != min\_num[6] 41

Cập nhật min\_num[4] = min(min\_num[4], min\_num[6]) = 1

Cập nhật 42

3. Kiểm tra num và min\_num của 4

num[4] == min\_num[4] num[4] != min\_num[4] 43

Cập nhật min\_num[3] = min(min\_num[3], min\_num[4]) = 1

Cập nhật 44

3. Kiểm tra num và min\_num của 3

num[3] == min\_num[3] num[3] != min\_num[3] 45

Lấy các đỉnh trong ngăn xếp ra cho đến khi lấy được 3

Các đỉnh được lấy ra: 6,4,7,5,2,3

Nội dung còn lại của ngăn xếp:

Cập nhật 46

Về các thành phần liên thông

Help Clear shift Delete Edit Undo Red Black

Test	Got	
✓ Test tự động	<div>1. Kiểm tra DFS<ul style="list-style-type: none"><li>Bước 1. SCC(1): 1. Đỉnh số và đưa u vào ngăn xếp, với u = 1</li><li>Bước 2. SCC(1): 2. Xử lý các đỉnh kề của 1</li><li>Bước 3. SCC(1): 2a. Duyệt 2</li><li>Bước 4. SCC(2): 1. Đỉnh số và đưa u vào ngăn xếp, với u = 2</li><li>Bước 5. SCC(2): 2. Xử lý các đỉnh kề của 2</li><li>Bước 6. SCC(2): 2a. Duyệt 5</li><li>Bước 7. SCC(5): 1. Đỉnh số và đưa u vào ngăn xếp, với u = 5</li><li>Bước 8. SCC(5): 2. Xử lý các đỉnh kề của 5</li><li>Bước 9. SCC(5): 2a. Duyệt 7</li><li>Bước 10. SCC(7): 1. Đỉnh số và đưa u vào ngăn xếp, với u = 7</li><li>Bước 11. SCC(7): 2. Xử lý các đỉnh kề của 7</li><li>Bước 12. SCC(7): 2b. Cập nhật min_id của 2</li><li>Bước 13. SCC(7): 2b. Cập nhật min_id của 2</li><li>Bước 14. SCC(7): 2b. Cập nhật min_id của 3</li><li>Bước 15. SCC(7): 2b. Cập nhật min_id của 3</li><li>Bước 16. SCC(7): 3b. id[u] != min_id[u] với u = 7</li><li>Bước 17. SCC(5): 2a. Cập nhật min_id của 7</li><li>Bước 18. SCC(5): 3b. id[u] != min_id[u] với u = 5</li><li>Bước 19. SCC(2): 2a. Cập nhật min_id của 5</li><li>Bước 20. SCC(2): 3b. id[u] != min_id[u] với u = 2</li><li>Bước 21. SCC(1): 2a. Cập nhật min_id của 2</li><li>Bước 22. SCC(1): 2a. Duyệt 4</li><li>Bước 23. SCC(4): 1. Đỉnh số và đưa u vào ngăn xếp, với u = 4</li><li>Bước 24. SCC(4): 2. Xử lý các đỉnh kề của 4</li><li>Bước 25. SCC(4): 2a. Duyệt 1</li><li>Bước 26. SCC(1): 1. Đỉnh số và đưa u vào ngăn xếp, với u = 1</li><li>Bước 27. SCC(1): 2. Xử lý các đỉnh kề của 1</li><li>Bước 28. SCC(1): 3a. id[u] == min_id[u] với u = 1</li><li>Bước 29. SCC(1): 3a. Tìm được SCC của 1</li><li>Bước 30. SCC(4): 2a. Cập nhật min_id của 1</li><li>Bước 31. SCC(4): 2b. Cập nhật min_id của 3</li><li>Bước 32. SCC(4): 2b. Cập nhật min_id của 3</li><li>Bước 33. SCC(4): 2b. Cập nhật min_id của 5</li><li>Bước 34. SCC(4): 2b. Cập nhật min_id của 5</li><li>Bước 35. SCC(4): 2a. Duyệt 6</li><li>Bước 36. SCC(6): 1. Đỉnh số và đưa u vào ngăn xếp, với u = 6</li><li>Bước 37. SCC(6): 2. Xử lý các đỉnh kề của 6</li><li>Bước 38. SCC(6): 2a. Bỏ qua đỉnh 1</li></ul></div>	✓