

Algoritmi e strutture dati

Algoritmo di Tarjan per Strongly Connected Components

Alberto Montresor

Università di Trento

2016/08/11

This work is licensed under a Creative Commons
Attribution-ShareAlike 4.0 International License.



Sommario

- 1 Descrizione algoritmo

Algoritmo

SET *scc*(GRAPH *G*)

int *current* $\leftarrow 1$

int[] *index* \leftarrow **new int**[1...*G.n*] $\leftarrow \{0, \dots, 0\}$

int[] *lowlink* \leftarrow **new int**[1...*G.n*]

boolean[] *inStack* \leftarrow **new boolean**[1...*G.n*]

STACK *stack* \leftarrow Stack()

SET *components* \leftarrow Set()

foreach $u \in G.V()$ **do**

if *index*[*u*] = 0 **then**
 strongconnect(*G*, *u*)

Algoritmo

strongconnect(GRAPH g , NODE u)

$index[u] \leftarrow lowlink[u] \leftarrow current$

$current \leftarrow current + 1$

$stack.push(u)$

$inStack[u] \leftarrow \mathbf{true}$

foreach $v \in G.adj(u)$ **do**

if $index[v] = 0$ **then**

strongconnect(G, v)

$lowlink[u] = \min(lowlink[u], lowlink[v])$

else if $inStack[v]$ **then**

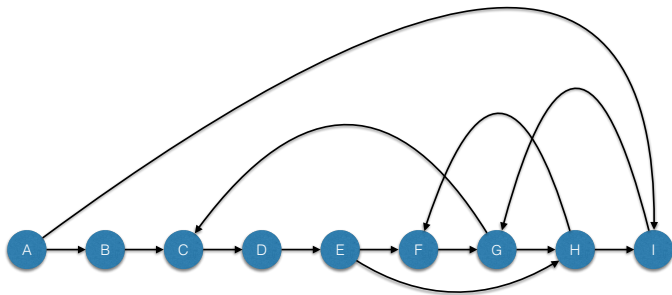
$lowlink[u] = \min(lowlink[u], lowlink[v])$

if $lowlink[u] = index[u]$ **then**

 SET $component \leftarrow Set()$

repeat

Esempio di funzionamento



Index

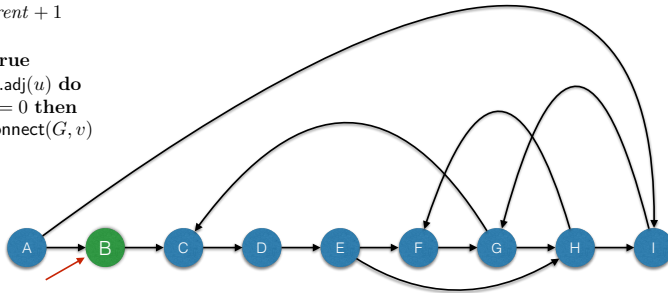
LowLink

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```



Index 1

LowLink 1

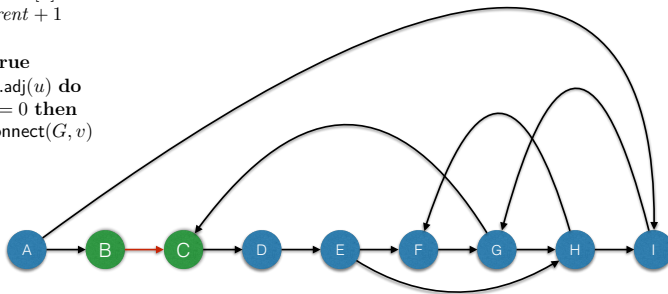
Stack B

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```



Index	1	2
-------	---	---

LowLink	1	2
---------	---	---

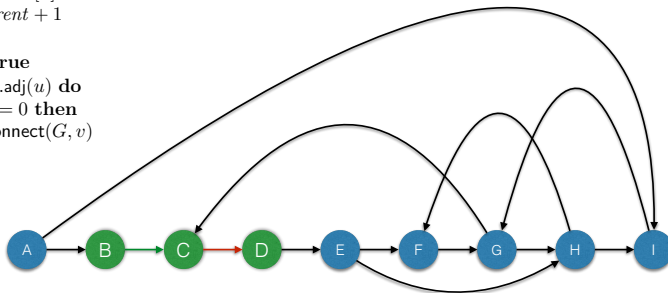
Stack	B	C
-------	---	---

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```



Index		1	2	3
-------	--	---	---	---

LowLink		1	2	3
---------	--	---	---	---

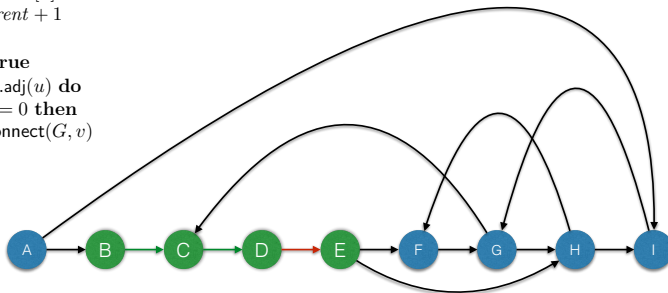
Stack		B	C	D
-------	--	---	---	---

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```



Index		1	2	3	4			
-------	--	---	---	---	---	--	--	--

LowLink		1	2	3	4			
---------	--	---	---	---	---	--	--	--

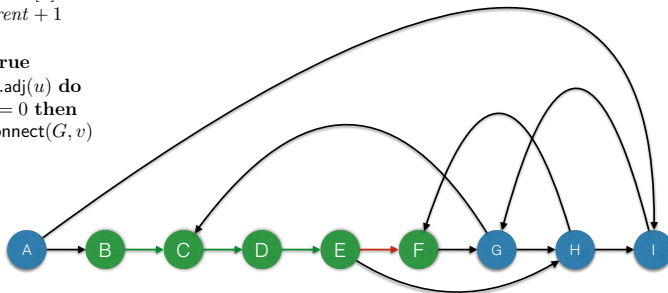
Stack		B	C	D	E			
-------	--	---	---	---	---	--	--	--

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```



Index		1	2	3	4	5
-------	--	---	---	---	---	---

LowLink		1	2	3	4	5
---------	--	---	---	---	---	---

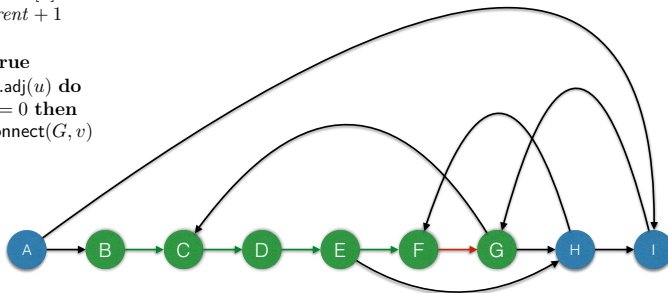
Stack		B	C	D	E	F
-------	--	---	---	---	---	---

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```



Index

1

2

3

4

5

6

LowLink

1

2

3

4

5

6

Stack

B

C

D

E

F

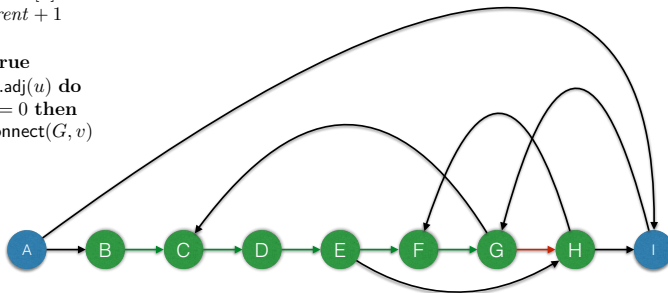
G

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```



Index		1	2	3	4	5	6	7
-------	--	---	---	---	---	---	---	---

LowLink		1	2	3	4	5	6	7
---------	--	---	---	---	---	---	---	---

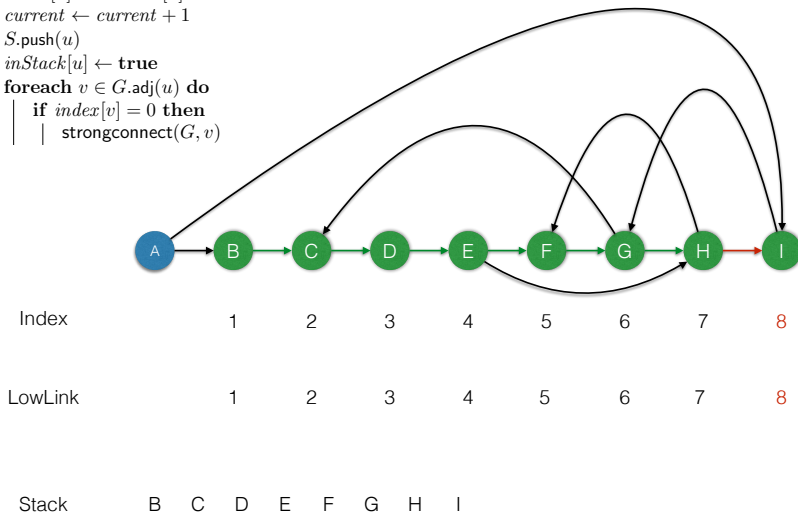
Stack	B	C	D	E	F	G	H
-------	---	---	---	---	---	---	---

Esempio di funzionamento

```

index[u] ← lowlink[u] ← current
current ← current + 1
S.push(u)
inStack[u] ← true
foreach v ∈ G.adj(u) do
  if index[v] = 0 then
    strongconnect(G, v)

```

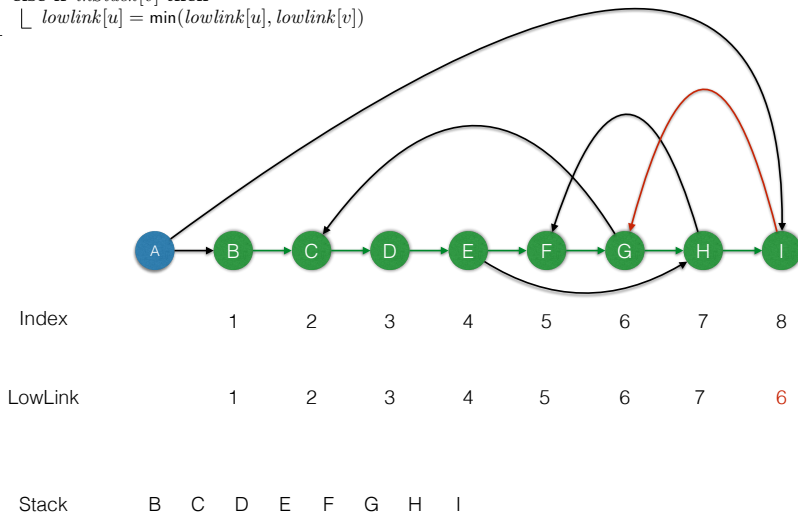


Esempio di funzionamento

```

else if inStack[v] then
   $\lfloor \text{lowlink}[u] = \min(\text{lowlink}[u], \text{lowlink}[v])$ 

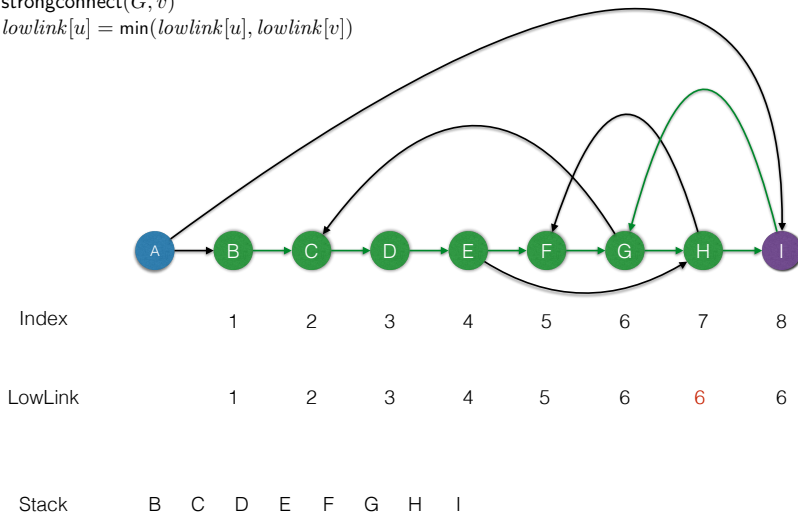
```



Esempio di funzionamento

`strongconnect(G, v)`

$lowlink[u] = \min(lowlink[u], lowlink[v])$

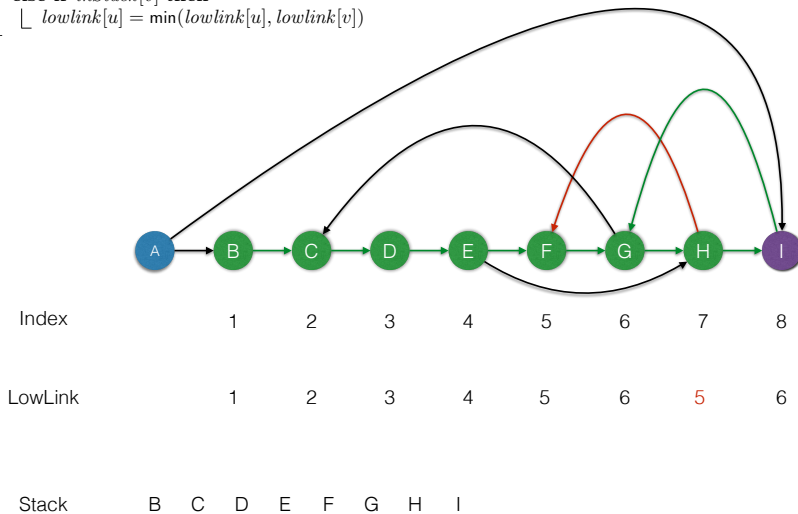


Esempio di funzionamento

```

else if inStack[v] then
   $\lfloor \text{lowlink}[u] = \min(\text{lowlink}[u], \text{lowlink}[v])$ 

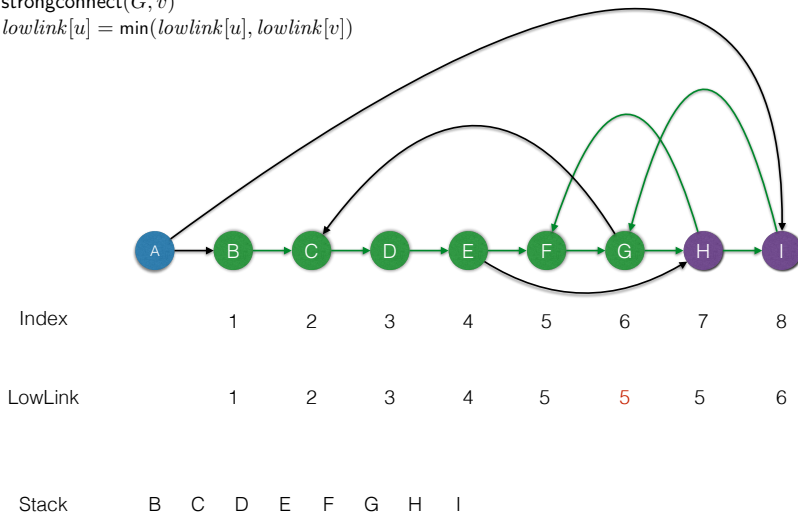
```



Esempio di funzionamento

`strongconnect(G, v)`

$lowlink[u] = \min(lowlink[u], lowlink[v])$

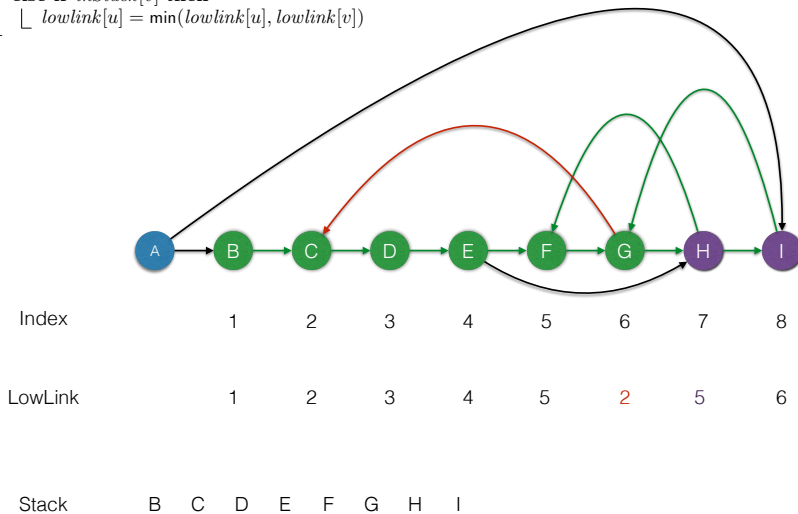


Esempio di funzionamento

```

else if inStack[v] then
   $\lfloor \text{lowlink}[u] = \min(\text{lowlink}[u], \text{lowlink}[v])$ 

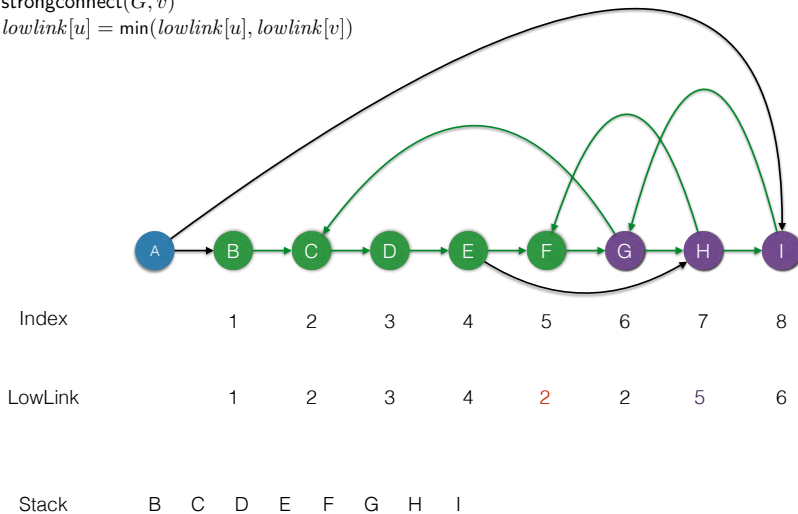
```



Esempio di funzionamento

`strongconnect(G, v)`

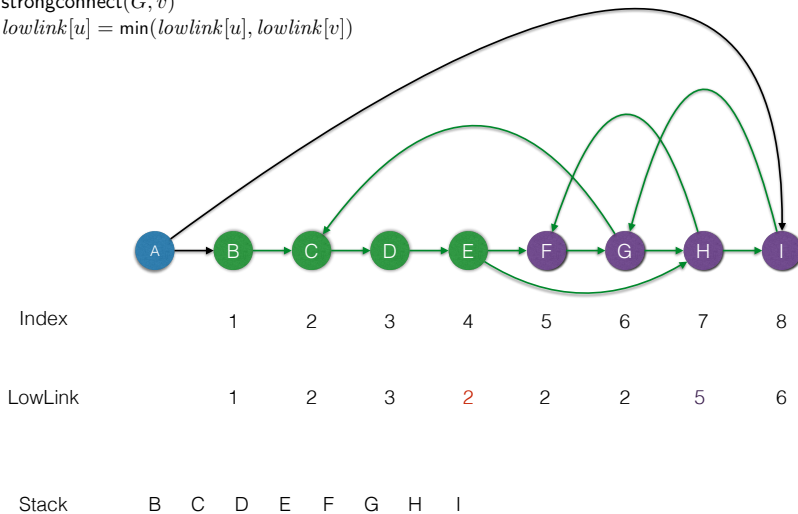
$lowlink[u] = \min(lowlink[u], lowlink[v])$



Esempio di funzionamento

`strongconnect(G, v)`

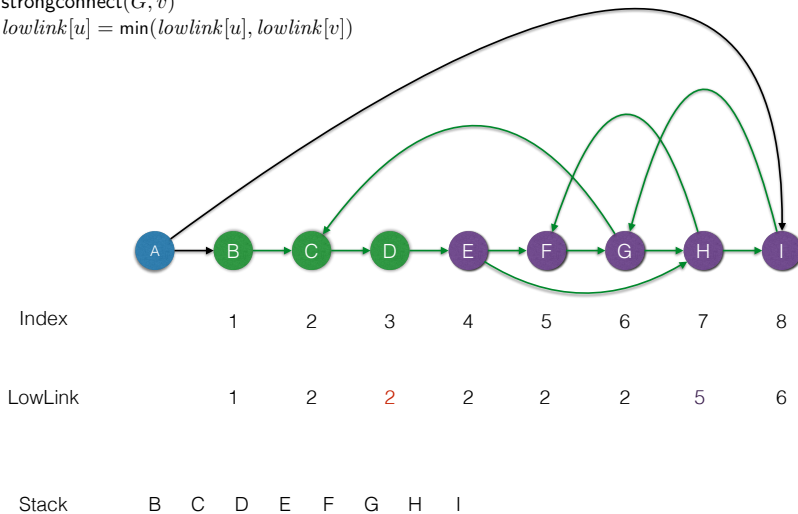
$lowlink[u] = \min(lowlink[u], lowlink[v])$



Esempio di funzionamento

`strongconnect(G, v)`

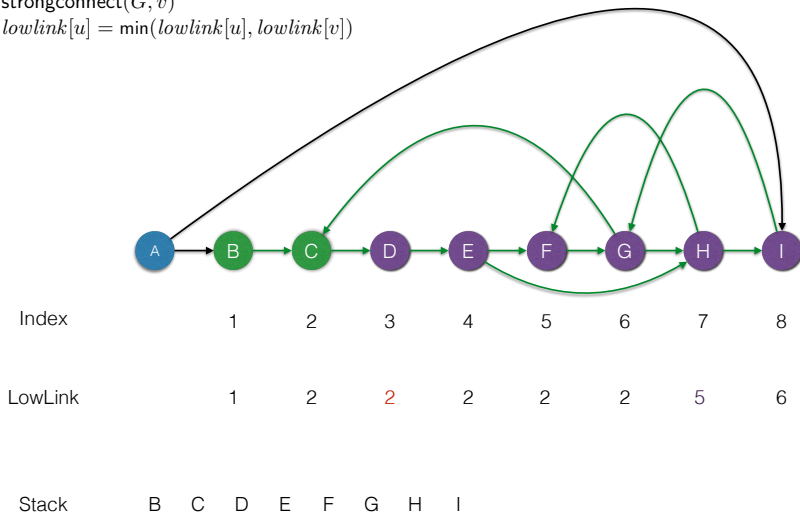
$lowlink[u] = \min(lowlink[u], lowlink[v])$



Esempio di funzionamento

`strongconnect(G, v)`

$lowlink[u] = \min(lowlink[u], lowlink[v])$

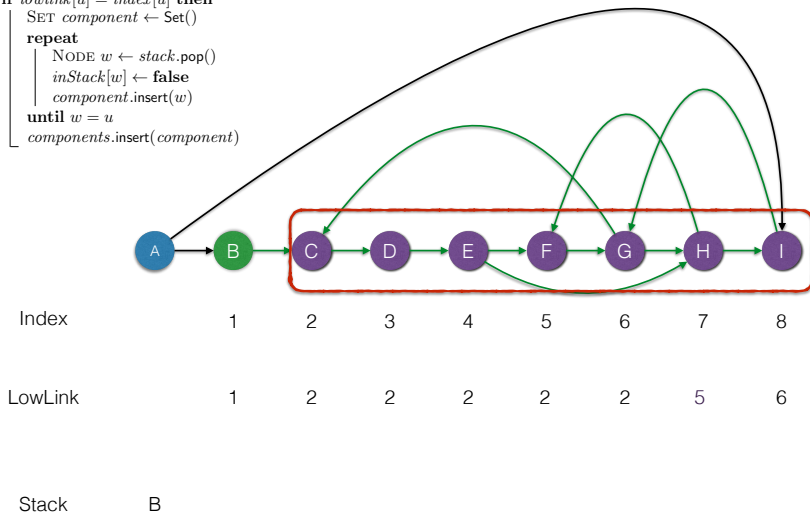


Esempio di funzionamento

```

if  $lowlink[u] = index[u]$  then
  SET  $component \leftarrow Set()$ 
  repeat
    NODE  $w \leftarrow stack.pop()$ 
     $inStack[w] \leftarrow \text{false}$ 
     $component.insert(w)$ 
  until  $w = u$ 
   $components.insert(component)$ 

```

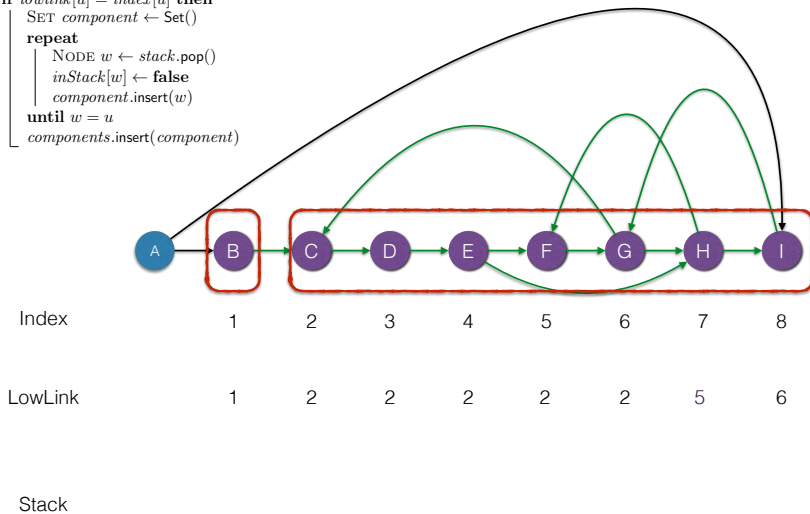


Esempio di funzionamento

```

if  $lowlink[u] = index[u]$  then
  SET  $component \leftarrow Set()$ 
  repeat
    NODE  $w \leftarrow stack.pop()$ 
     $inStack[w] \leftarrow false$ 
     $component.insert(w)$ 
  until  $w = u$ 
   $components.insert(component)$ 

```



Esempio di funzionamento

