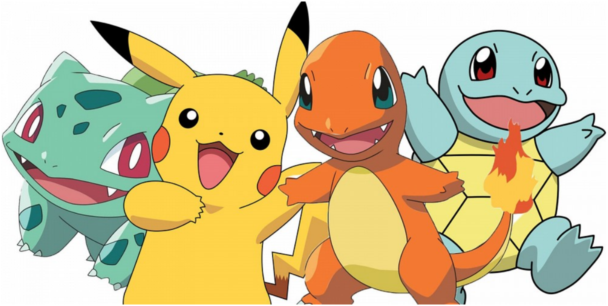
Soluzione Progetto 1 ASD a.a. 2016/2017



Pokémon Park

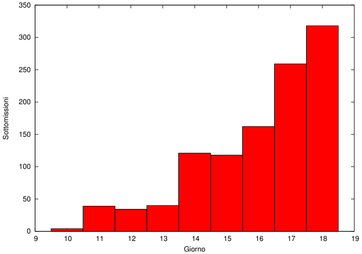
Alessio Guerrieri e Cristian Consonni

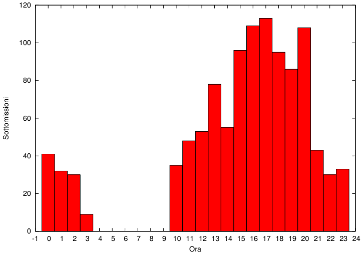
November 21, 2016

Statistiche

Statistiche

Numero sottoposizioni: 1095





- 89 gruppi iscritti (l'ultimo alle **23:27 del 07/11**);
- 179 studenti;
- $\sim$ 17 ore di ricevimento (compresi i laboratori);
- 59 mail ricevute;

Risultati

Punteggi (classifica completa sul sito)

- $P < 30$ $\longrightarrow$ progetto non passato
- $30 \leq P < 75$ $\longrightarrow$ 1 punto bonus (23 gruppi)
- $75 \leq P < 95$ $\longrightarrow$ 2 punti bonus (26 gruppi)
- $95 \leq P \leq 100$ $\longrightarrow$ 3 punti bonus (26 gruppi)

Soluzione dei casi con un solo ciclo

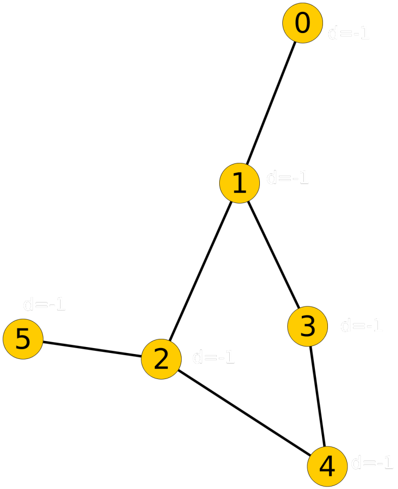
- Individuiamo il ciclo e la sua lunghezza $\rightarrow K$;
- Usiamo una visita in profondità e marchiamo ogni nodo visitato con la sua profondità rispetto alla sorgente;
- Se incontriamo un nodo già visitato, che non sia il nodo da cui proveniamo (**padre** del nodo corrente rispetto alla visita):
  - $\Rightarrow$ **ciclo**
- La dimensione del ciclo è data dalla differenza di profondità fra i nodi collegati dall'arco che chiude il ciclo;
- La visita va eseguita su tutti i nodi non visitati per tenere conto del caso in cui ci siano più componenti connesse.

$\Rightarrow$ soluzione: `unciclo.cpp`, 42 SLOC (*Source Lines Of Code*)

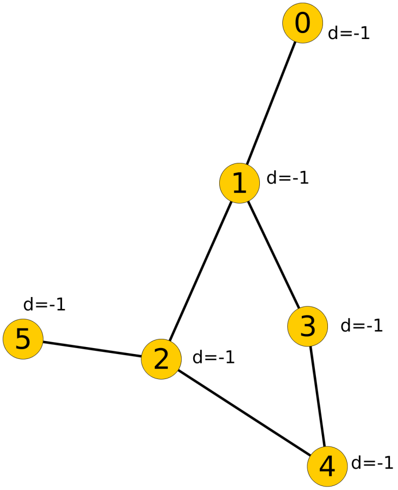
$\Rightarrow$ complessità: $\Omega(V + E)$

$\Rightarrow$ 36 punti

Esempio con un solo ciclo

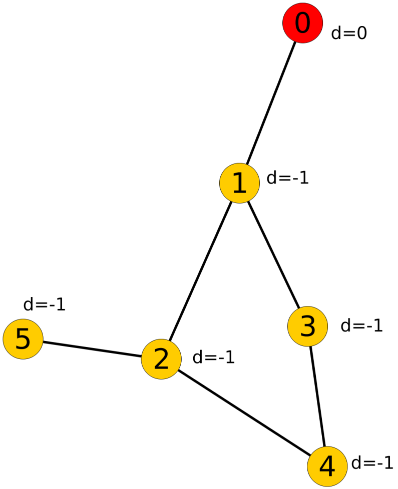


Esempio con un solo ciclo



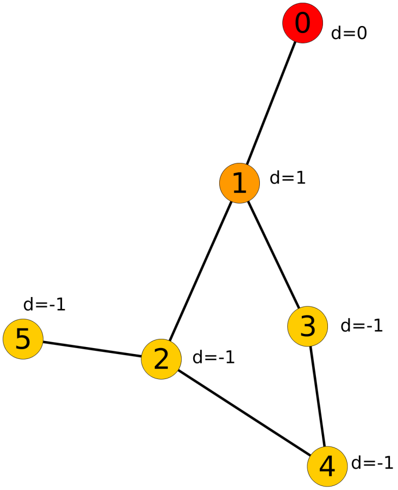
- nodi a profondità -1

Esempio con un solo ciclo



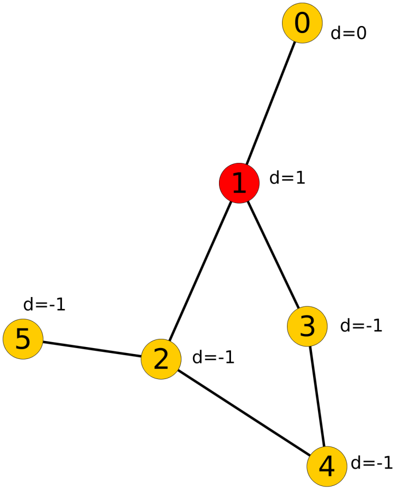
- nodi a profondità -1
- start, visito 0, (profondità 0)

Esempio con un solo ciclo



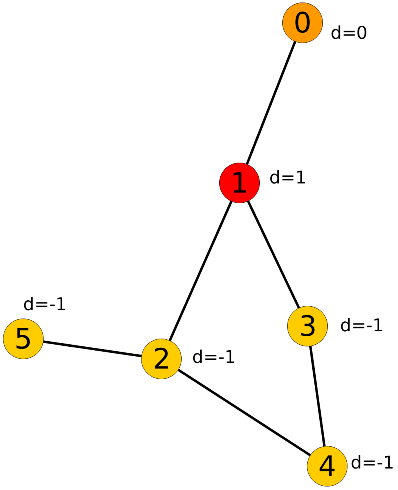
- nodi a profondità -1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)

Esempio con un solo ciclo



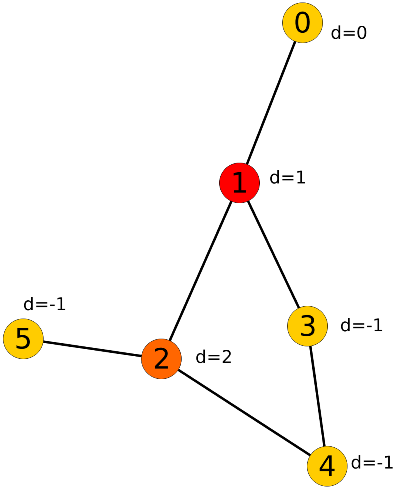
- nodi a profondità -1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1

Esempio con un solo ciclo



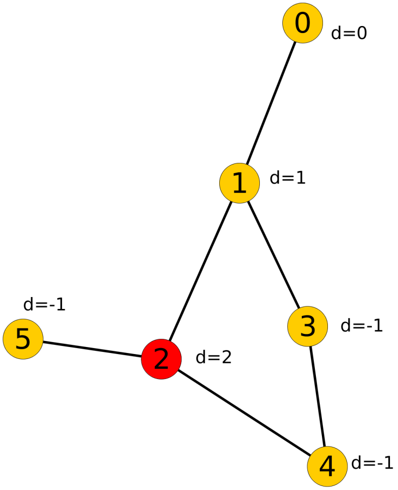
- nodi a profondità -1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 $\rightarrow$ ignoro)

Esempio con un solo ciclo



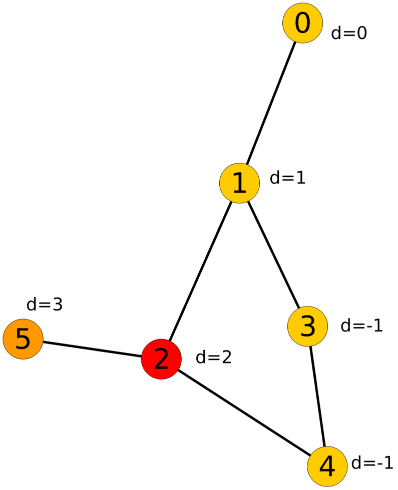
- nodi a profondità -1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 $\rightarrow$ ignoro)
- vicino 2, (profondità 2)

Esempio con un solo ciclo



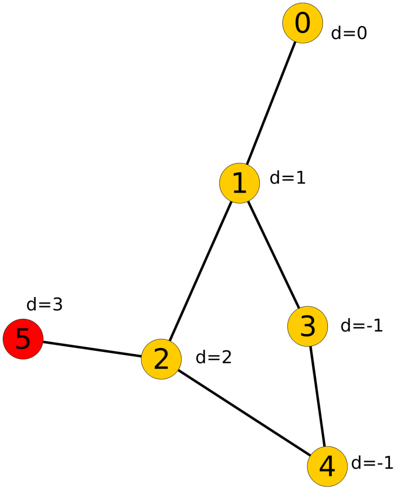
- nodi a profondità -1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 $\rightarrow$ ignoro)
- vicino 2, (profondità 2)
- visito 2

Esempio con un solo ciclo



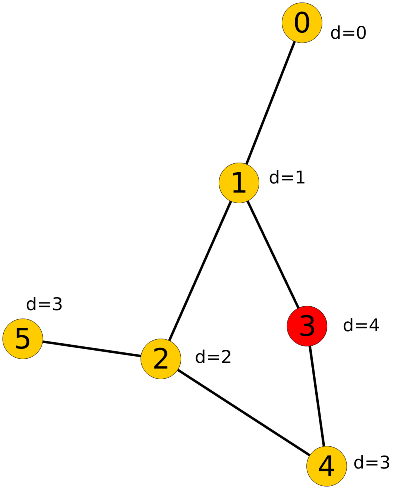
- nodi a profondità -1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 $\rightarrow$ ignoro)
- vicino 2, (profondità 2)
- visito 2
- vicino 5, (profondità 3)

Esempio con un solo ciclo



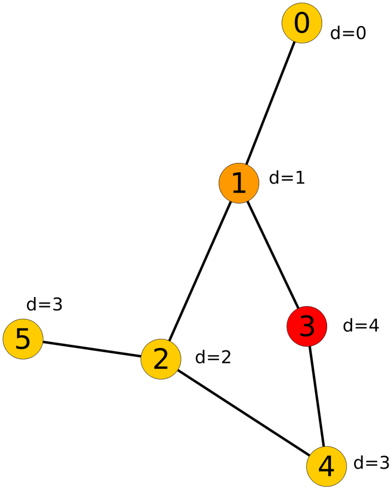
- nodi a profondità -1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 $\rightarrow$ ignoro)
- vicino 2, (profondità 2)
- visito 2
- vicino 5, (profondità 3)
- visito 5

Esempio con un solo ciclo



- visito 5
- ...
- visito 3

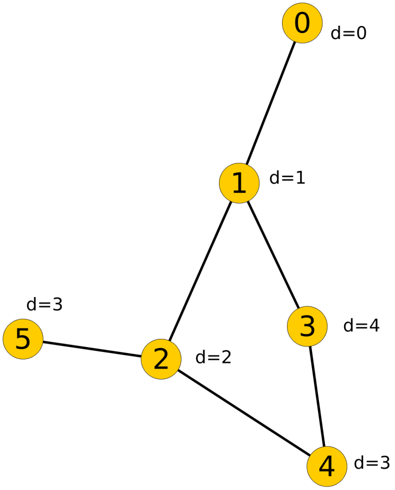
Esempio con un solo ciclo



- visito 5
- ...
- visito 3
- vicino 1, $\texttt{1.d} \neq -1$, $\texttt{1.d} < \texttt{3.d}$

$\Rightarrow$ ciclo

Esempio con un solo ciclo



- visito 5
- ...
- visito 3
- vicino 1, $\texttt{1.d} \neq -1$, $\texttt{1.d} < \texttt{3.d}$

$\Rightarrow$ ciclo

$\Rightarrow$ lunghezza ciclo $4 - 1 + 1 = 4$

Soluzione dei casi con più cicli semplici disgiunti

- Individuiamo la lunghezza di ogni ciclo con l'algoritmo precedente;
- K è il MCD delle lunghezze dei cicli;

$\Rightarrow$ soluzione: `cicli.cpp`, 52 SLOC

$\Rightarrow$ complessità: $\Omega(V + E)$

$\Rightarrow$ 51 punti

Soluzione dei casi con più cicli semplici disgiunti con posizionamento ($o_1 + o_2$)

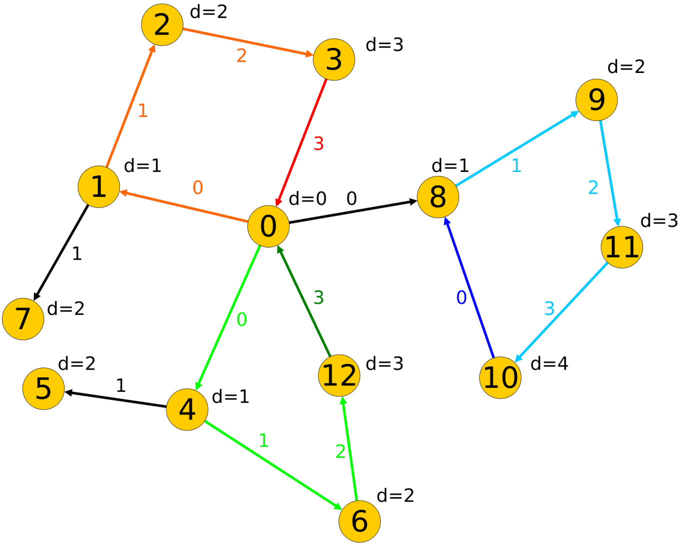
- Individuiamo K con l'algoritmo precedente;
- Dato un nodo di profondità P, assegnamo ai suoi archi uscenti il Pokémon ($P \bmod K$);
- L'orientamento degli archi è ottenuto seguendo la visita:
  - agli archi che chiudono i cicli assegnamo la profondità del nodo più basso, $\max(p.depth, a.depth)$;
  - agli altri archi assegnamo la profondità del nodo più alto, $\min(p.depth, a.depth)$;

$\Rightarrow$ soluzione: `cicli_con_posizionamento.cpp`, 67 SLOC

$\Rightarrow$ complessità: $\Omega(V + E)$

$\Rightarrow$ 85 punti

Esempio con più cicli semplici disgiunti e posizionamento ($o_1 + o_2$)

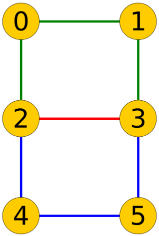


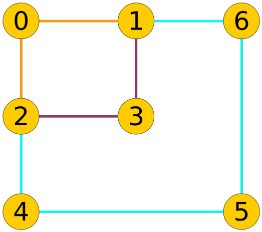
Soluzione del caso generico: intuizione

Intuizione

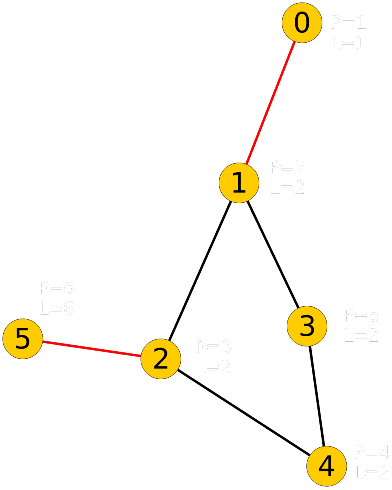
- Trovare i gruppi di strade che, se rimosse, **“rompono”** gli stessi cicli.
- Trovare i gruppi di strade per i quali, qualora si percorra una strada del gruppo, allora si devono percorrere tutte le altre strade di quel gruppo.

$\Rightarrow$ ogni gruppo deve rispettare i requisiti, K è il MCD del numero di elementi di ogni gruppo.





Soluzione del caso generico: bridges



Definizione

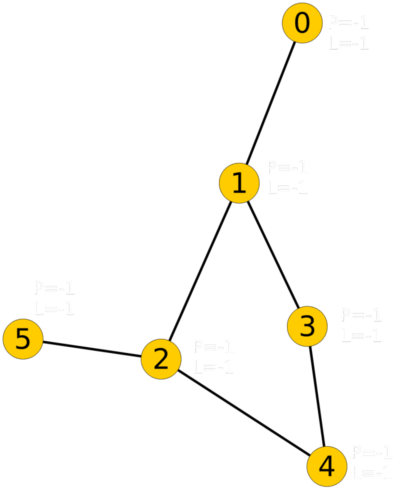
- Definiamo **bridges** gli archi la cui rimozione rende disconnesso il grafo;
- (nota) Archi che collegano nodi che non appartengono ad alcun ciclo sono bridges;

Soluzione del caso generico: algoritmo di Tarjan

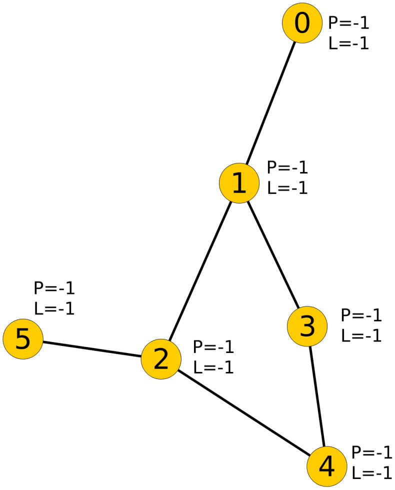
Si esegue una visita in profondità ricorsiva in pre-order:

- in ogni nodo del grafo vengono mantenute due variabili:
  - `pre`: contatore che viene incrementato secondo il numero di chiamate ricorsive della visita DFS;
  - `low`: specifica il livello del nodo meno profondo raggiungibile per ogni nodo;
- ad ogni passo della visita si incrementano `pre` e `low`;
- quando un ciclo si chiude, il backtracking della DFS trasmette `low` a tutti i nodi appartnenti al ciclo;
- archi uscenti (nel verso della visita) che portano a nodi per i quali `pre` = `low` sono bridges;

Esempio di esecuzione dell'algoritmo di Tarjan

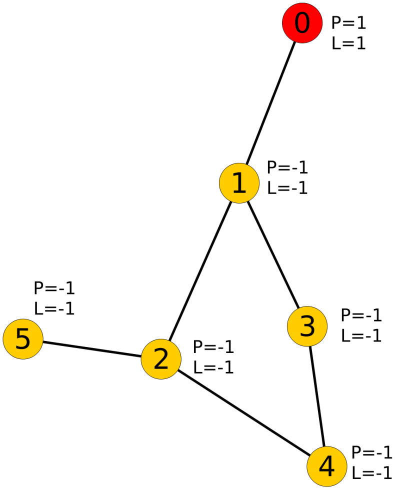


Esempio di esecuzione dell'algoritmo di Tarjan



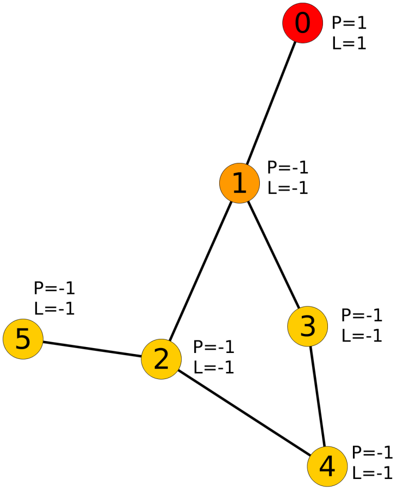
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$

Esempio di esecuzione dell'algoritmo di Tarjan



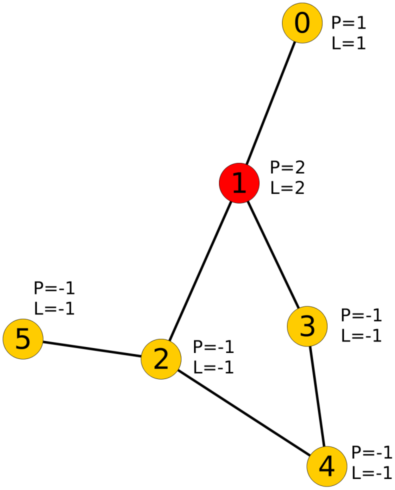
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$
- start, visito 0 $(\texttt{count} = 1)$ $\rightarrow$ $(\texttt{pre} = 1,\ \texttt{low} = 1)$

Esempio di esecuzione dell'algoritmo di Tarjan



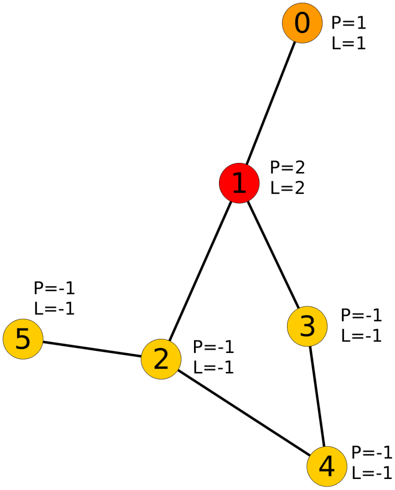
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$
- start, visito 0 $(\texttt{count} = 1)$ $\rightarrow (\texttt{pre} = 1, \texttt{low} = 1)$
- vicino 1

Esempio di esecuzione dell'algoritmo di Tarjan



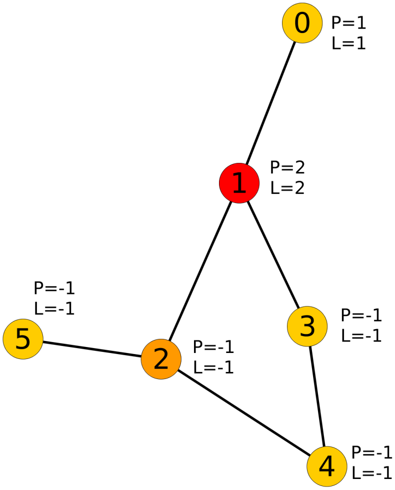
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$
- start, visito 0 ($\texttt{count} = 1$) $\rightarrow$ ($\texttt{pre} = 1$, $\texttt{low} = 1$)
- vicino 1
- visito 1 ($\texttt{count} = 2$) $\rightarrow$ ($\texttt{pre} = 2$, $\texttt{low} = 2$)

Esempio di esecuzione dell'algoritmo di Tarjan



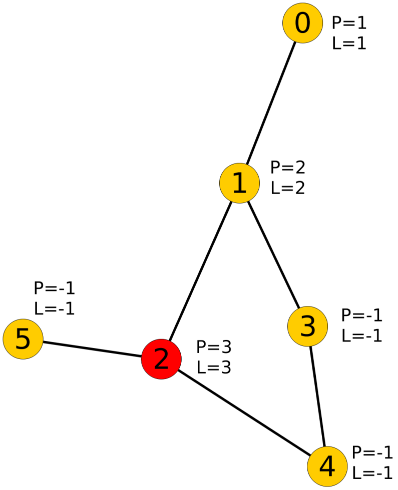
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$
- start, visito 0 ($\texttt{count} = 1$) $\longrightarrow$ ($\texttt{pre} = 1$, $\texttt{low} = 1$)
- vicino 1
- visito 1 ($\texttt{count} = 2$) $\longrightarrow$ ($\texttt{pre} = 2$, $\texttt{low} = 2$)
- vicino 0 (padre di 1 $\rightarrow$ ignoro)

Esempio di esecuzione dell'algoritmo di Tarjan



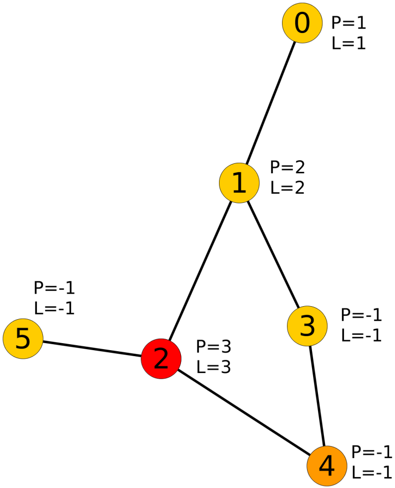
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$
- start, visito 0 $(\texttt{count} = 1)$ $\rightarrow$ $(\texttt{pre} = 1, \texttt{low} = 1)$
- vicino 1
- visito 1 $(\texttt{count} = 2)$ $\rightarrow$ $(\texttt{pre} = 2, \texttt{low} = 2)$
- vicino 0 (padre di 1 $\rightarrow$ ignoro)
- vicino 2

Esempio di esecuzione dell'algoritmo di Tarjan



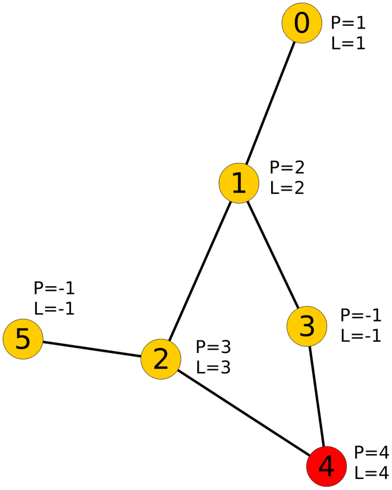
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$
- start, visito 0 ($\texttt{count} = 1$) $\longrightarrow$ ($\texttt{pre} = 1$, $\texttt{low} = 1$)
- vicino 1
- visito 1 ($\texttt{count} = 2$) $\longrightarrow$ ($\texttt{pre} = 2$, $\texttt{low} = 2$)
- vicino 0 (padre di 1 $\rightarrow$ ignoro)
- vicino 2
- visito 2 ($\texttt{count} = 3$) $\longrightarrow$ ($\texttt{pre} = 3$, $\texttt{low} = 3$)

Esempio di esecuzione dell'algoritmo di Tarjan



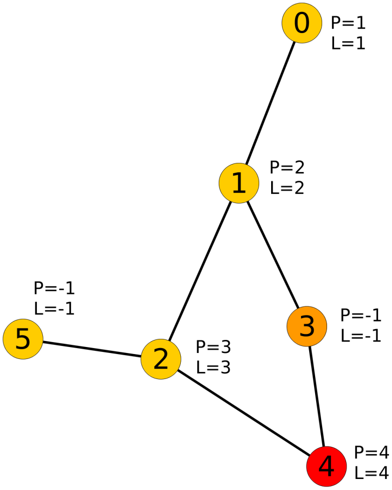
- tutti i nodi hanno $\texttt{pre} = -1$, $\texttt{low} = -1$
- start, visito 0 ($\texttt{count} = 1$) $\longrightarrow$ ($\texttt{pre} = 1$, $\texttt{low} = 1$)
- vicino 1
- visito 1 ($\texttt{count} = 2$) $\longrightarrow$ ($\texttt{pre} = 2$, $\texttt{low} = 2$)
- vicino 0 (padre di 1 $\rightarrow$ ignoro)
- vicino 2
- visito 2 ($\texttt{count} = 3$) $\longrightarrow$ ($\texttt{pre} = 3$, $\texttt{low} = 3$)
- vicino 4

Esempio di esecuzione dell'algoritmo di Tarjan



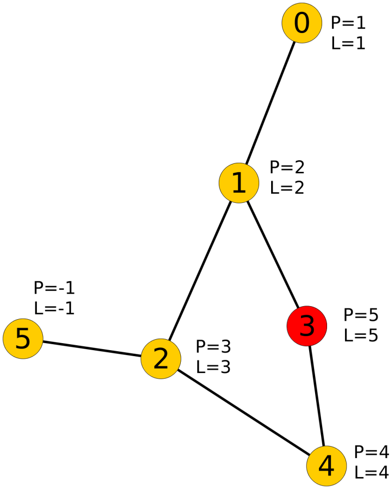
- ...
- visito 2 ($\texttt{count} = 3$)
  $\rightarrow$ ($\texttt{pre} = 3$, $\texttt{low} = 3$)
- vicino 4
- visito 4 ($\texttt{count} = 4$)
  $\rightarrow$ ($\texttt{pre} = 4$, $\texttt{low} = 4$)

Esempio di esecuzione dell'algoritmo di Tarjan



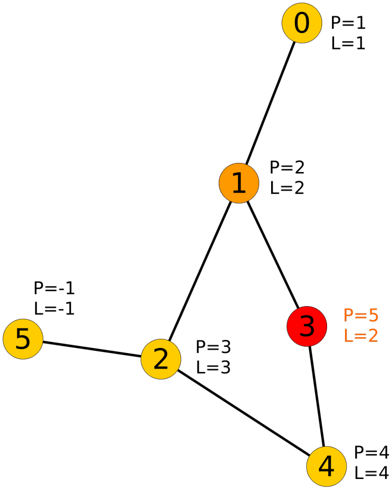
- ...
- visito 2 ($\texttt{count} = 3$)
  $\rightarrow$ ($\texttt{pre} = 3$, $\texttt{low} = 3$)
- vicino 4
- visito 4 ($\texttt{count} = 4$)
  $\rightarrow$ ($\texttt{pre} = 4$, $\texttt{low} = 4$)
- vicino 3

Esempio di esecuzione dell'algoritmo di Tarjan



- ...
- visito 2 ($\texttt{count} = 3$)
  $\rightarrow$ ($\texttt{pre} = 3$, $\texttt{low} = 3$)
- vicino 4
- visito 4 ($\texttt{count} = 4$)
  $\rightarrow$ ($\texttt{pre} = 4$, $\texttt{low} = 4$)
- vicino 3
- visito 3 ($\texttt{count} = 5$)
  $\rightarrow$ ($\texttt{pre} = 5$, $\texttt{low} = 5$)

Esempio di esecuzione dell'algoritmo di Tarjan

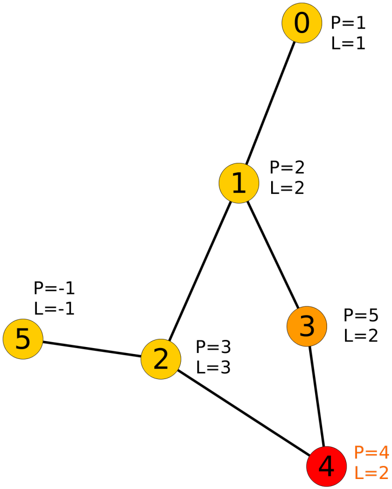


- . . .
- visito 2 $(\texttt{count} = 3)$
  $\rightarrow (\texttt{pre} = 3, \texttt{low} = 3)$
- vicino 4
- visito 4 $(\texttt{count} = 4)$
  $\rightarrow (\texttt{pre} = 4, \texttt{low} = 4)$
- vicino 3
- visito 3 $(\texttt{count} = 5)$
  $\rightarrow (\texttt{pre} = 5, \texttt{low} = 5)$

$\Rightarrow$ ciclo

$\Rightarrow$ $\texttt{vic.low} \rightarrow \texttt{n.low}$

Esempio di esecuzione dell'algoritmo di Tarjan



- . . .
- visito 3 (`count` $= 5$)

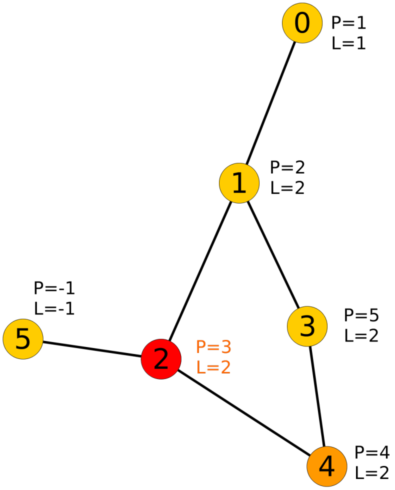
  $\longrightarrow$ (`pre` $= 5$, `low` $= 5$)

$\Rightarrow$ ciclo

$\Rightarrow$ `vic.low` $\rightarrow$ `n.low`

- backtrack (`3.low` $\rightarrow$ `4.low`)

Esempio di esecuzione dell'algoritmo di Tarjan



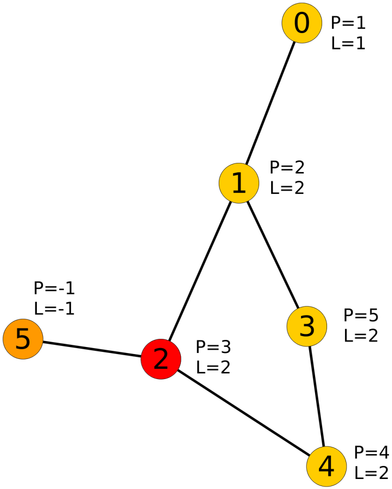
- ...
- visito 3 (`count` = 5)
  $\rightarrow$ (`pre` = 5, `low` = 5)

$\Rightarrow$ ciclo

$\Rightarrow$ `vic.low` $\rightarrow$ `n.low`

- backtrack (`3.low` $\rightarrow$ `4.low`)
- backtrack (`4.low` $\rightarrow$ `2.low`)

Esempio di esecuzione dell'algoritmo di Tarjan

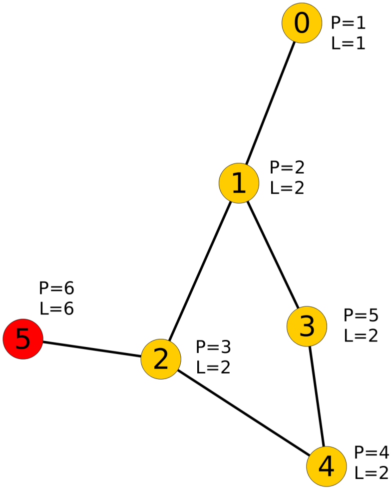


$\Rightarrow$ ciclo

$\Rightarrow$ `vic.low` $\rightarrow$ `n.low`

- backtrack (`3.low` $\rightarrow$ `4.low`)
- backtrack (`4.low` $\rightarrow$ `2.low`)
- vicino 5

Esempio di esecuzione dell'algoritmo di Tarjan



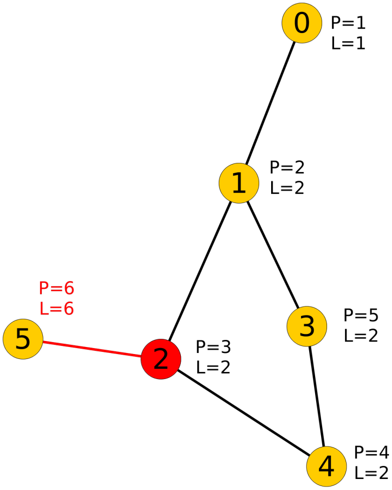
$\Rightarrow$ ciclo

$\Rightarrow$ `vic.low` $\rightarrow$ `n.low`

- backtrack (`3.low` $\rightarrow$ `4.low`)
- backtrack (`4.low` $\rightarrow$ `2.low`)
- vicino 5
- visito 5 (`count` $= 6$)

  $\rightarrow$ (`pre` $= 6$, `low` $= 6$)

Esempio di esecuzione dell'algoritmo di Tarjan

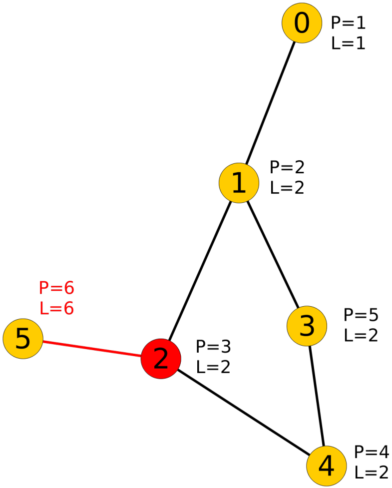


$\Rightarrow$ ciclo

$\Rightarrow$ `vic.low` $\rightarrow$ `n.low`

- backtrack (`3.low` $\rightarrow$ `4.low`)
- backtrack (`4.low` $\rightarrow$ `2.low`)
- vicino 5
- visito 5 (count $= 6$)
  $\rightarrow$ (pre $= 6$, low $= 6$)
- backtrack

Esempio di esecuzione dell'algoritmo di Tarjan

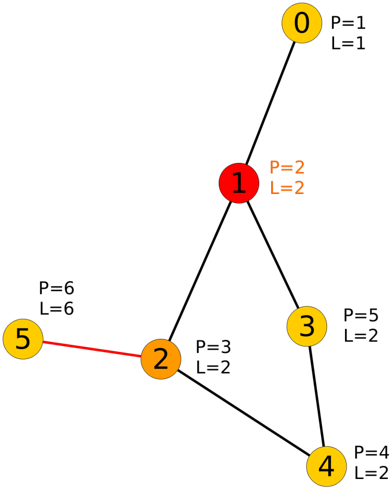


$\Rightarrow$ ciclo

$\Rightarrow$ `vic.low` $\to$ `n.low`

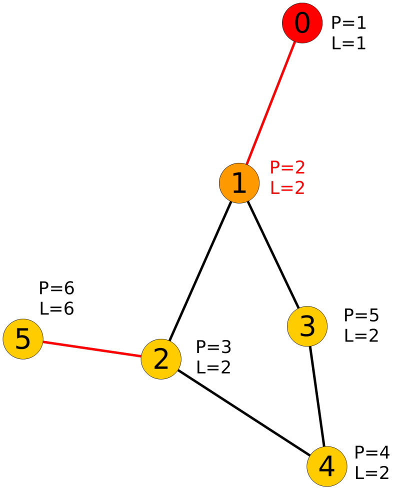
- backtrack (`3.low` $\to$ `4.low`)
- backtrack (`4.low` $\to$ `2.low`)
- vicino 5
- visito 5 (`count` $= 6$)
  $\to$ (`pre` $= 6$, `low` $= 6$)
- backtrack
  `pre` $=$ `low` $\Rightarrow$ **bridge**

Esempio di esecuzione dell'algoritmo di Tarjan



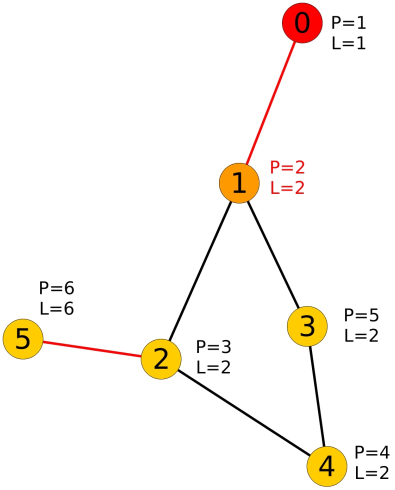
- backtrack ($\texttt{3.low} \rightarrow \texttt{4.low}$)
- backtrack ($\texttt{4.low} \rightarrow \texttt{2.low}$)
- vicino 5
- visito 5 ($\texttt{count} = 6$)
  $\rightarrow$ ($\texttt{pre} = 6$, $\texttt{low} = 6$)
- backtrack
  $\texttt{pre} = \texttt{low} \Rightarrow$ **bridge**
- backtrack ($\texttt{2.low} \rightarrow \texttt{1.low}$)

Esempio di esecuzione dell'algoritmo di Tarjan



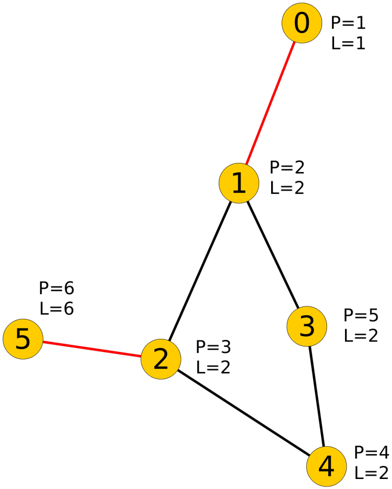
- backtrack ($3.\texttt{low} \to 4.\texttt{low}$)
- backtrack ($4.\texttt{low} \to 2.\texttt{low}$)
- vicino 5
- visito 5 ($\texttt{count} = 6$)
  $\to$ ($\texttt{pre} = 6$, $\texttt{low} = 6$)
- backtrack
  $\texttt{pre} = \texttt{low} \Rightarrow$ **bridge**
- backtrack ($2.\texttt{low} \to 1.\texttt{low}$)
- backtrack

Esempio di esecuzione dell'algoritmo di Tarjan



- backtrack $(\texttt{3.low} \to \texttt{4.low})$
- backtrack $(\texttt{4.low} \to \texttt{2.low})$
- vicino 5
- visito 5 $(\texttt{count} = 6)$
  $\to (\texttt{pre} = 6,\ \texttt{low} = 6)$
- backtrack
  $\texttt{pre} = \texttt{low} \Rightarrow$ **bridge**
- backtrack $(\texttt{2.low} \to \texttt{1.low})$
- backtrack
  $\texttt{pre} = \texttt{low} \Rightarrow$ **bridge**

Esempio di esecuzione dell'algoritmo di Tarjan



- fine

Soluzione del caso generico: descrizione

- Con una prima passata di Tarjan si tolgono tutti gli archi che non appartengono ad alcun ciclo;
- Per ogni arco non-bridge a, segniamo a come da ignorare ed usiamo Tarjan sul grafo rimanente per trovare gli archi che appartengono allo stesso gruppo di a;
- K è il MCD delle grandezze dei gruppi trovati nel punto precedente;
- Si scorre ogni gruppo di archi e si numerano archi dello stesso gruppo progessivamente $\pmod{K}$.

⇒ soluzione: `generica.cpp`, 122 SLOC

⇒ complessità: $\Omega\left(E \cdot (V + E)\right)$

⇒ 100 punti

Note finali

- Classifiche e sorgenti sul sito (controllate i numeri di matricola):
  - http://judge.science.unitn.it/slides/asd16/classifica_prog1.pdf
  - Assumiamo gli stessi gruppi, in caso di cambiamenti scrivetemi (cristian.consonni@unitn.it)
- Chi ha passato il primo progetto non è obbligato a fare il secondo.