

ARTIKULACE A MOSTY

Petr Ryšavý

19. září 2023

Katedra počítačů, FEL, ČVUT

ARTIKULACE

Definice (Artikulace) *Artikulace* v grafu G je vrchol, jehož odstraněním se v grafu G zvýší počet slabě souvislých komponent.

- Hledáme tzv. *single-point-of-failure* body
- Např. switch v síti, uzlová stanice na železnici

- Který algoritmus nalezne komponenty slabé souvislosti?

- Který algoritmus nalezne komponenty slabé souvislosti?
- Můžeme spouštět DFS pro graf G bez každého jednotlivého vrcholu.
- $\mathcal{O}(|V| \cdot (|V| + |E|))$

- Pokud má vrchol DFS stromu alespoň dva nepropojené podstromy, pak jde o artikulaci

- Pokud má vrchol DFS stromu alespoň dva nepropojené podstromy, pak jde o artikulaci
- Pro každé volání DFS počítáme počet navštívených podgrafů
- Při hledání potomka jdeme pouze po nenavštívených uzlech
- Zvýšíme počítadlo pro každé volání DFS

- Vrchol u je artikulací, pokud má potomka, ze kterého nevede hrana zpět do nějakého předka u

- Vrchol u je artikulací, pokud má potomka, ze kterého nevede hrana zpět do nějakého předka u
- Potřebujeme časové značky, kdy vrchol otevřeme
- Používáme low-link, podobně jako v Tarjanově algoritmu
- Hrana, kterou jsme se dostali do uzlu z rodiče není zpětná!

Pseudokód

```
function ARTICULATION-POINTS(graph)
  disc, low  $\leftarrow$  empty 1-D arrays; time  $\leftarrow$  -1
  for all node  $\in$  graph do
    if UNVISITED(node) then
      DEPTH-FIRST-SEARCH(graph, node, null )
    end if
  end for
end function

function DEPTH-FIRST-SEARCH(graph, u, parent)
  children  $\leftarrow$  0; MARK-VISITED(u)
  disc[u]  $\leftarrow$  low[u]  $\leftarrow$  ++ time
  for all edges (u, v) do
    if UNVISITED(v) then
      children  $\leftarrow$  children + 1
      DEPTH-FIRST-SEARCH(graph, v, u)
      low[u]  $\leftarrow$  min{low[u], low[v]}
      if parent  $\neq$  null  $\wedge$  low[v]  $\geq$  disc[u] then
        MARK-ARTICULATION(u)
      end if
    else if v  $\neq$  parent then low[u]  $\leftarrow$  min{low[u], disc[v]}
    end if
  end for
  if parent == null  $\wedge$  children > 1 then
    MARK-ARTICULATION(u)
  end if
end function
```


MOSTY

Definice (Most) *Most* v grafu G je hrana, jejímž odstraněním z grafu G se zvýší počet slabě souvislých komponent grafu G .

- Obdoba artikulací pro hrany.
- Opět hledáme single-connection-of-failure
- Např. riziková spojení v ethernetové síti, silniční síti apod.

- Hrana $\{u, v\}$ je mostem, pokud z u nevede cesta do nějakého rodiče vrcholu v
- Hranu $\{u, v\}$ za tuto cestu nepočítáme!

- Opět použijeme čas otevření a low-link
- Aby hrana $\{u, v\}$ (expandovaná z u do v) byla mostem, musí být low-link vrcholu v větší než index vrcholu u .

```
function BRIDGES(graph)
  disc, low, parent  $\leftarrow$  empty 1-D arrays
  time  $\leftarrow$  -1
  for all node  $\in$  graph do
    if UNVISITED(node) then
      DEPTH-FIRST-SEARCH(graph, node)
    end if
  end for
end function

function DEPTH-FIRST-SEARCH(graph, u)
  MARK-VISITED(u)
  disc[u]  $\leftarrow$  low[u]  $\leftarrow$  ++ time
  for all edges (u, v) do
    if UNVISITED(v) then
      parent[v] = u
      DEPTH-FIRST-SEARCH(graph, v)
      low[u]  $\leftarrow$  min{low[u], low[v]}
      if low[v] > disc[u] then
        MARK-BRIDGE(u, v)
      end if
    else if v  $\neq$  parent[u] then low[u]  $\leftarrow$  min{low[u], disc[v]}
    end if
  end for
end function
```


- heavily inspired by Tim Roughgarden's online courses,
<http://theory.stanford.edu/~tim/videos.html>
- Robert Sedgewick and Kevin Wayne, Algorithms,
<http://algs4.cs.princeton.edu/home/>, namely
<http://algs4.cs.princeton.edu/44sp/>
- Halim, S., Halim, F., Skiena, S. S., & Revilla, M. A. (2013).
Competitive Programming 3. Lulu Independent Publish.
- <https://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/>
- <https://www.geeksforgeeks.org/bridge-in-a-graph/>
- <https://www.youtube.com/watch?v=64KK9K4RpKE>
- <https://www.youtube.com/watch?v=Rhxs4k6DyMM>

DĚKUJI ZA POZORNOST.
ČAS NA OTÁZKY!