

BFS - Breath-First Search

- $stav[v] \in \{N\text{-neviděný, } O\text{-otevřený, } Z\text{-zavřený}\}$
- $comp, layer[v]$ číslo komponenty a vrstvy vrcholu v

Function $BFS(\text{graf } G)$

```
komponenty  $\leftarrow 0$ 
vrstva[*]  $\leftarrow \infty$ 
stav[*]  $\leftarrow N$ 
while  $\exists v : stav[v] = N$  do
  ++komponenty
  aktVrstva = 0
  seznam1  $\leftarrow \{v\}$ , seznam2  $\leftarrow \emptyset$ 
  while  $w \in seznam1$  do
    stav[w]  $\leftarrow Z$ 
    vrstva[w]  $\leftarrow$  aktVrstva
    for  $w$  soused  $v$  &  $stav[w] = N$  do
      přidej  $w$  do seznam2
      stav[w]  $\leftarrow O$ 
    end
    if seznam1  $\neq \emptyset$  then
      ++aktVrstva
      swap seznam1, seznam2
    end
  end
end
end
```

Použití

- detekce komponent
 - strom nejkratších cest
 - test acykličnosti, bipartitnosti
-

DFS - Depth-First Search

- $stav[v] \in \{N\text{-neviděný}, O\text{-otevřený}, Z\text{-zavřený}\}$
- $in, out[v]$ čas vstupu a opuštění vrcholu v
- $low[v]$ (pro neorient. graf) nejnižší čas, kam se lze vrátit z (potomků) v

Function *DFS*(graf G)

```
clock  $\leftarrow$  0
low[*]  $\leftarrow$   $+\infty$ 
stav[*]  $\leftarrow$   $N$ 
while  $\exists v : stav[v] = N$  do
  | VISIT( $v, -$ )
end
end
```

Function *VISIT*(vrchol v , hrana e)

```
in[ $v$ ]  $\leftarrow$  clock++
low[ $v$ ]  $\leftarrow$   $\infty$ 
stav[ $v$ ]  $\leftarrow$   $O$ 
while  $\exists (vw) : (vw) \neq e$  do
  | if  $stav[w] = N$  then
    | VISIT( $w, vw$ )
    | low[ $v$ ]  $\leftarrow$   $\min\{low[v], low[w]\}$ 
  | end
  | else
    | low[ $v$ ]  $\leftarrow$   $\min\{low[v], in[w]\}$ 
  | end
end
out[ $v$ ]  $\leftarrow$  clock++
stav[ $v$ ]  $\leftarrow$   $Z$ 
end
```

Použití

- detekce mostů a artikulací
 - detekce komponent silné souvislosti a 2-souvislosti
 - test acykličnosti
 - sestrojení topologického pořadí
-