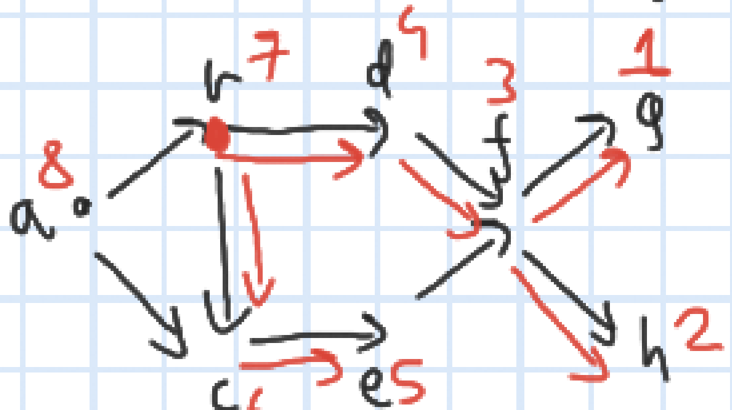
Algorytmy i Struktury Danych

Wykład 7

Zastosowania algorytmu DFS

1 Sortowanie topologiczne DAGu

- uruchamiamy DFS
- po przetworzeniu wierzchołka dopisujemy go na początku listy



a → b → c → e d → f → h g

2 Cykl Eulera

def Cykl Eulera w grafie G to taki cykl, który przez każdą krawędź G przechodzi dokładnie raz

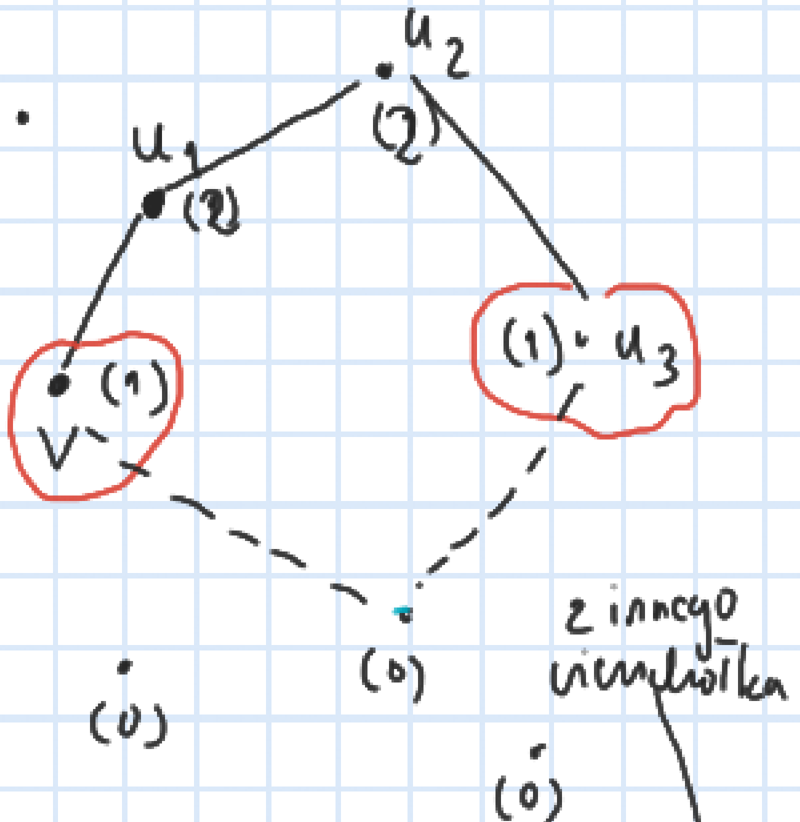
tw Graf nieskierowany i spójny posiada cykl Eulera wtw. gdy każdy jego wierzchołek ma parzysty stopień

"dowód"

Warunek konieczny: oczywiste

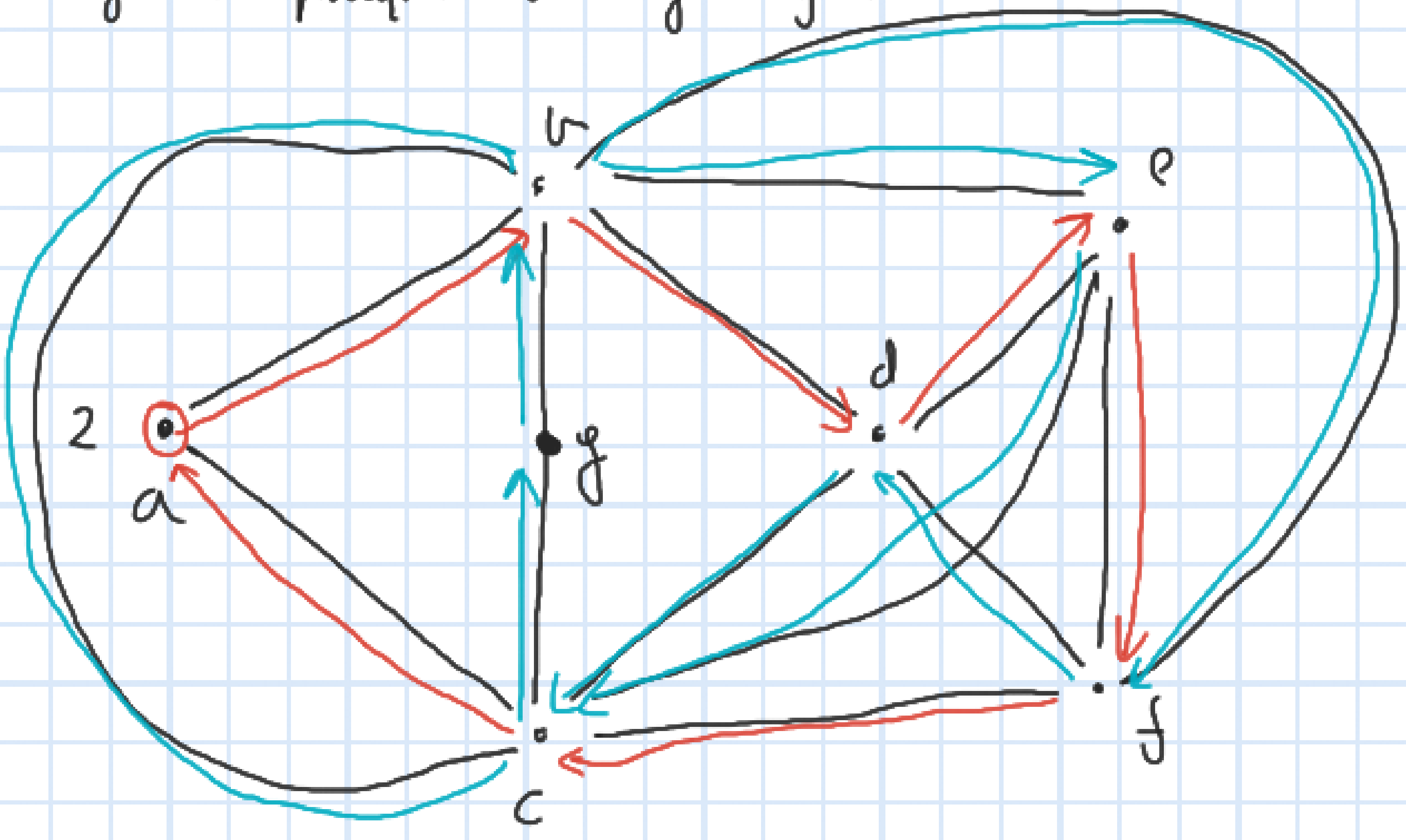
Warunek wystarczający:

- wybieramy dowolny wierzchołek v,
- startując z v wędrujemy, za każdym razem wybierając dowolną jeszcze nie użytą krawędź
- w końcu z powrotem dotrzemy do v
- albo znaleźliśmy cykl Eulera, albo możemy dokleić nowy podcykl startując



Algorytm

- wykonujemy DFS, ale:
  - po drodze "usuwamy" krawędzie po których przeszliśmy
  - do danego wierzchołka możemy wejść dowolną liczbę razy
- po przetworzeniu danego wierzchołka dopisujemy go na początku tworzonego cyklu



a b d e f c g b e c b f d c a

Złożoność

Taka sama jak DFS

$\underbrace{O(V+E)}_{\text{rep. listowa}}$, $\underbrace{O(V^2)}_{\text{rep. macierzowa}}$

łatwo wpaść w pułapkę złożoności

$O(V^2+VE)$, $O(V^3)$

---

def Cykl Hamiltona to cykl prosty, który odwiedza każdy wierzchołek dokładnie raz

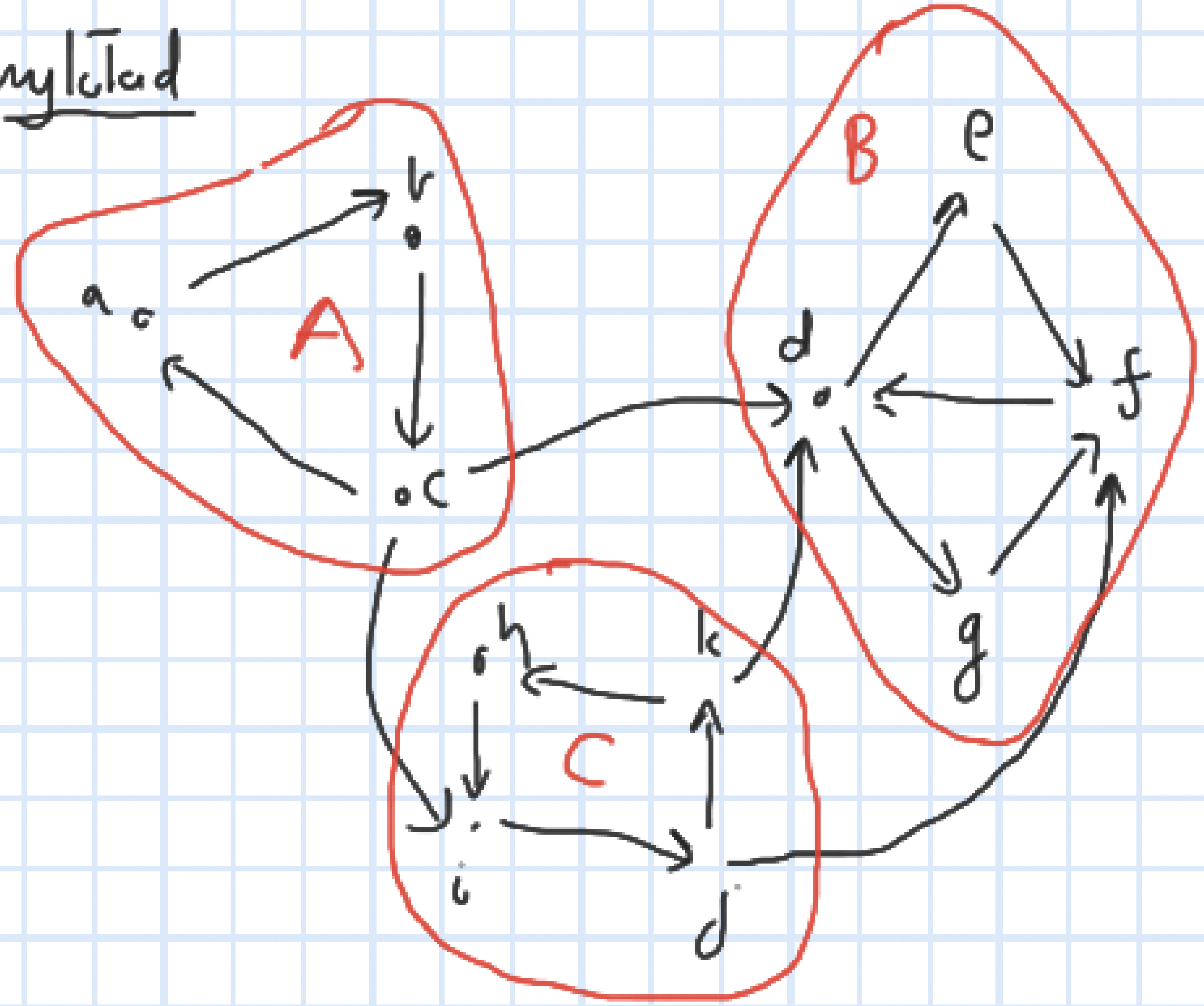
Algorytm: Brute-force — sprawdza $O(V!)$ możliwości

Nie da się wiele lepiej, bo rozpoznawanie czy graf ma cykl Hamiltona jest problemem NP-zupełnym

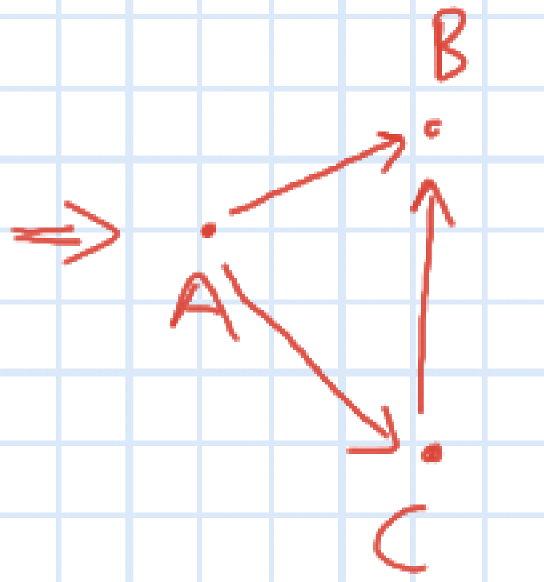
③ Silnie spójne składowe w grafie skierowanym

<u>def</u> Niech $G = (V, E)$ będzie pewnym grafem skierowanym. Mówimy, że $u, v \in V$ należą do tej samej silnie spójnej składowej jeśli istnieje ścieżka skierowana z u do v oraz z v do u

<u>Przykład</u>



$\Rightarrow$



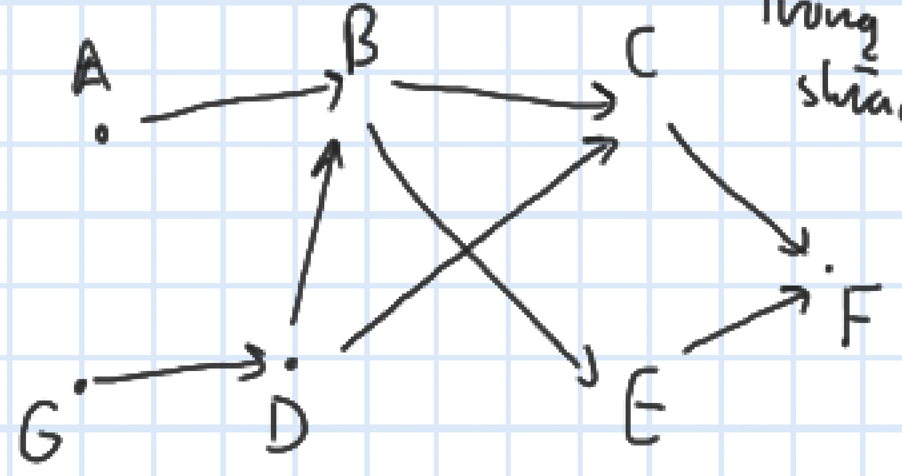
graf silnie spójnych składowych (DAG)

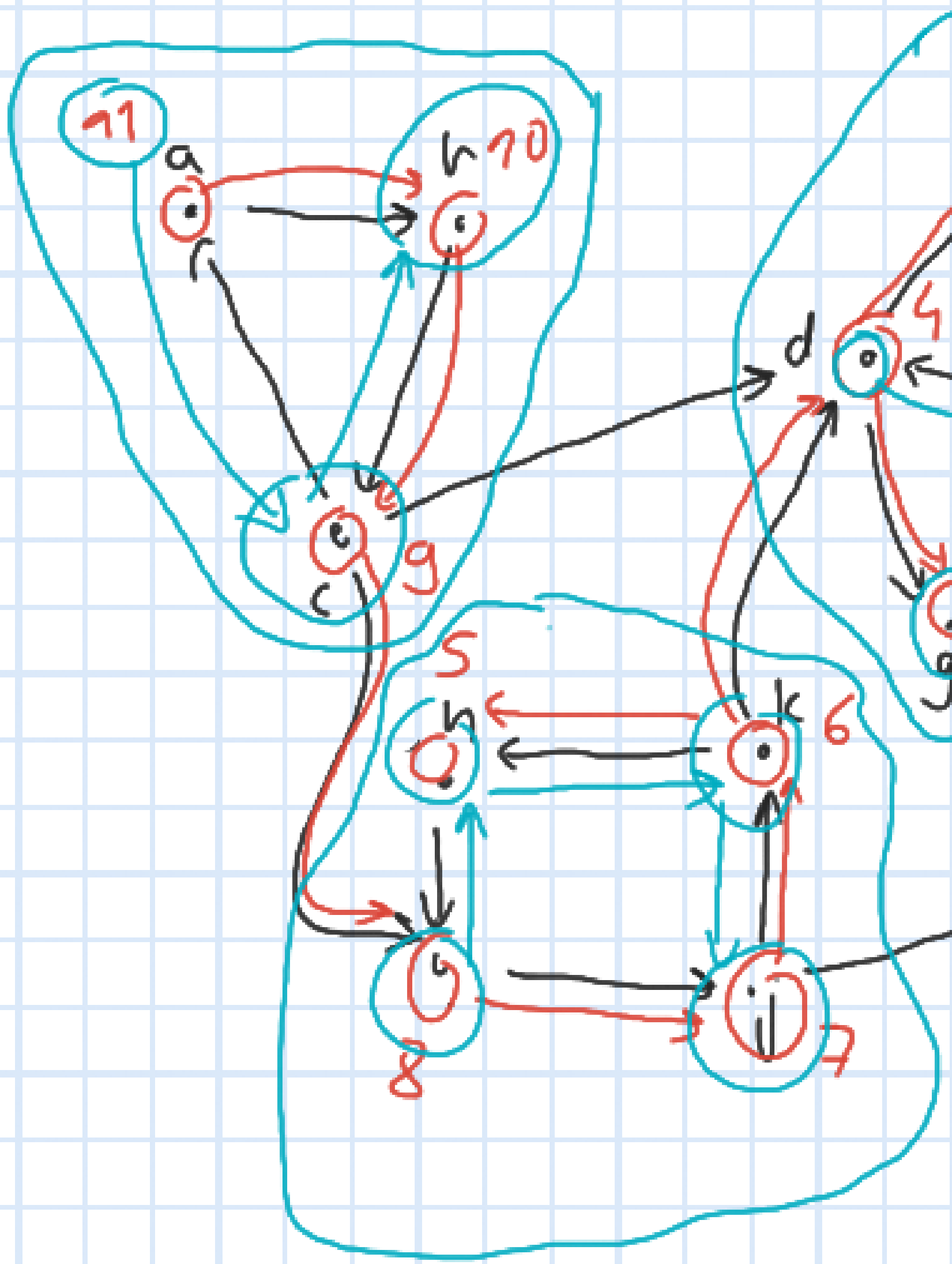
Algorytm

1. Wykonaj DFS na grafie G, zapisując czasy przetworzenia
2. Odwróć kierunek wszystkich krawędzi
3. Wykonaj DFS, wybierając startowe wierzchołki w kolejności malejących czasów przetworzenia z poprzedniego wykonania

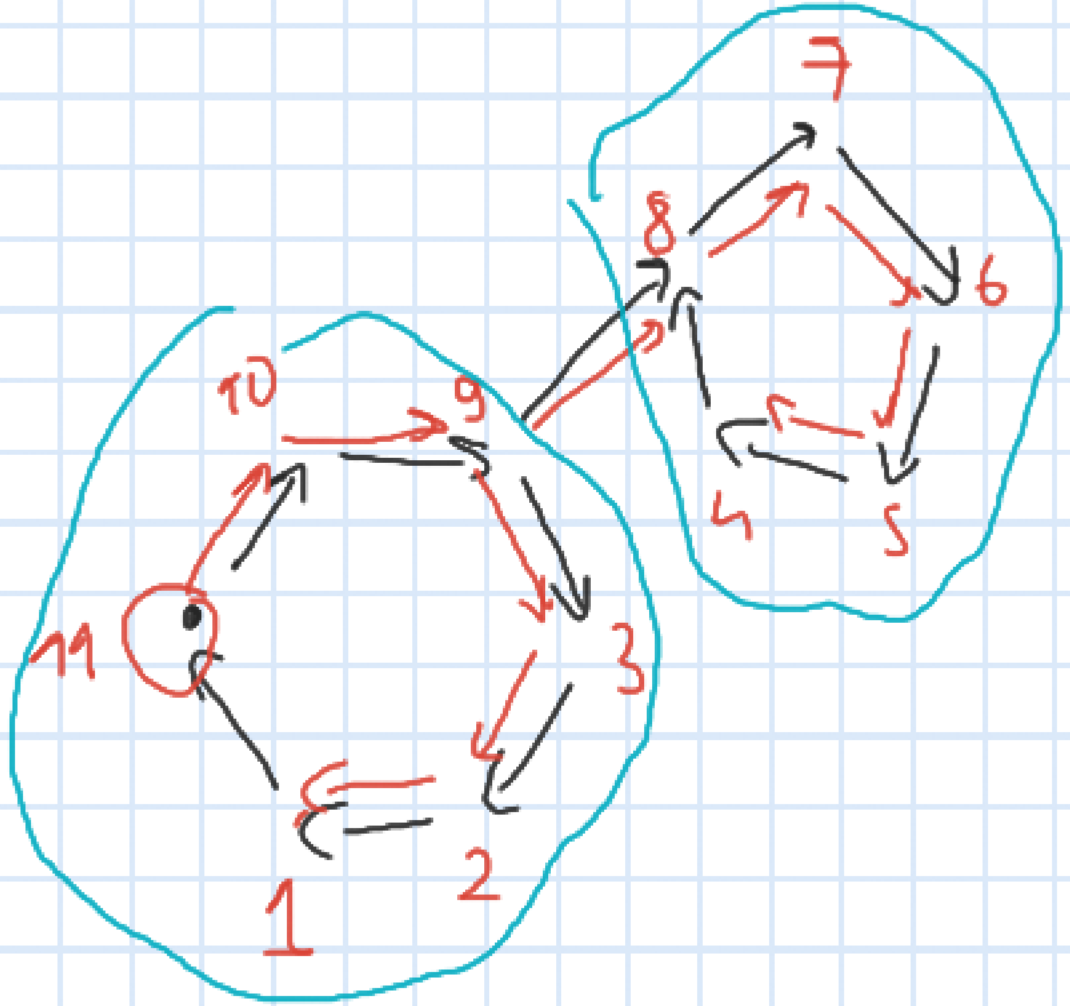
→ wierzchołki odwiedzone w danym "DFS Visit" tworzą silnie spójną składową

<u>Intuicja</u>





Przykład ↑



Złożoność

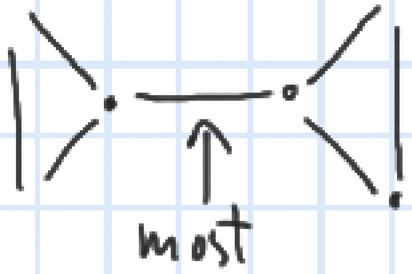
$O(V+E)$, $O(V^2)$

↑ listy

↑ macierz

4 Mosty w grafach nieskierowanych

<u>def</u> Krawędź e w spójnym grafie nieskierowanym G jest mostem jeśli jej usunięcie powoduje, że graf staje się niespójny



<u>tw</u> Krawędź e jest mostem wtw. gdy nie leży na żadnym cyklu prostym w grafie

<u>Dowód</u>

e jest mostem $\Rightarrow$ nie leży na cyklu
(bo wpp jej usunięcie nie rozspójnia grafu)

e nie leży na żadnym cyklu $\Rightarrow$ e jest mostem
(bo jeśli $e = \{u, v\}$, to jej usunięcie powoduje, że nie ma ścieżki z u do v)

Algorytm

1. Wykonaj DFS, dla każdego v zapisz jego czas odwiedzenia $d(v)$

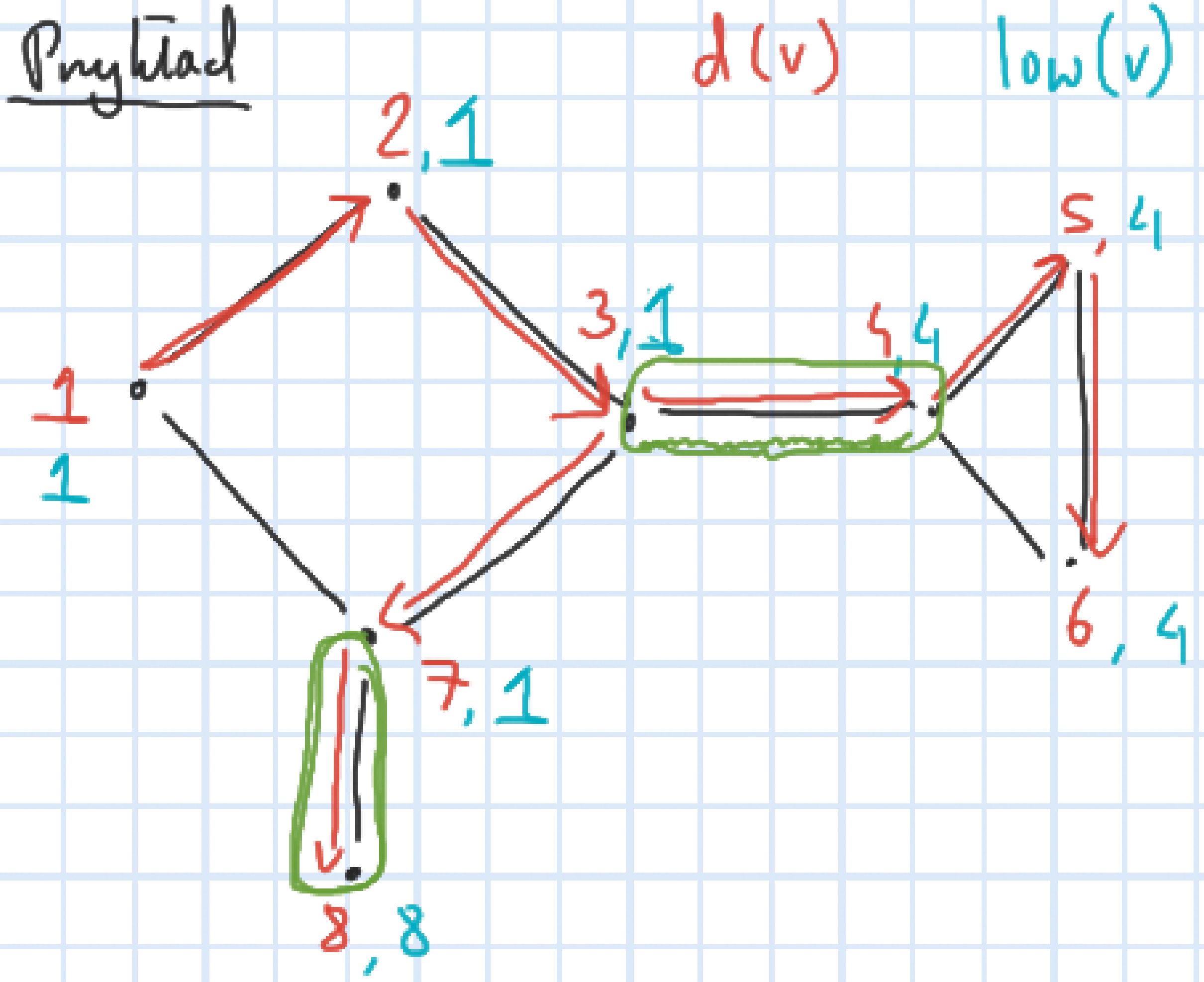
2. Dla każdego wierzchołka v oblicz

$$low(v) = \min\Big(d(v),\ \min_{\substack{\text{z } v \text{ jest krawędź} \\ \text{wsteczna do } u}} d(u),\ \min_{\substack{w \text{ to dziecko } v \\ \text{w drzewie DFS}}} (low(w)) \Big)$$

3. Mosty to krawędzie $\{v, \overset{\text{rodzic } v \text{ w drzewie DFS}}{p(v)}\}$ takie gdzie $d(v) = low(v)$

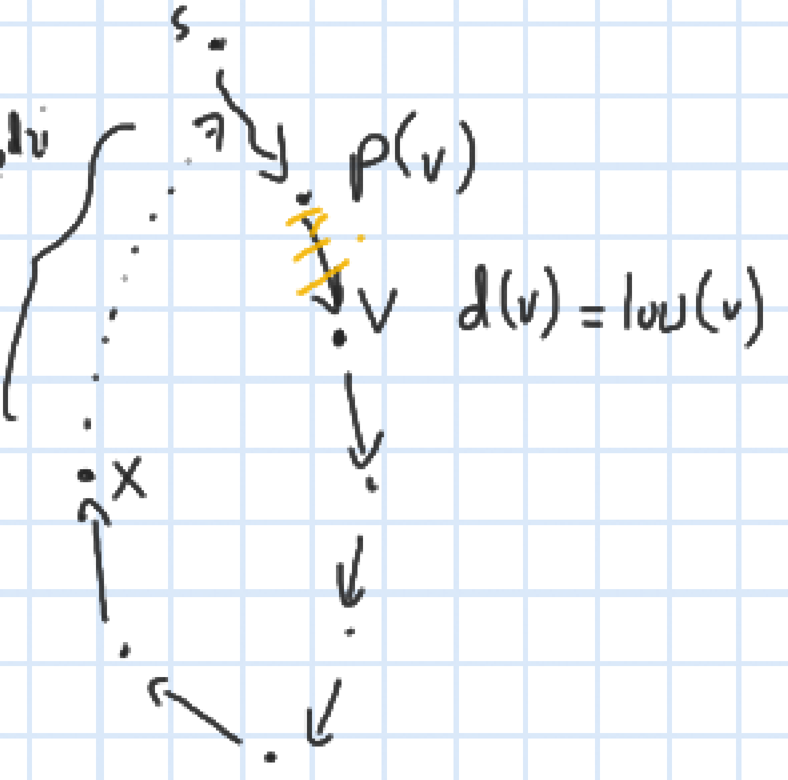
Idea: low(v) - identyfikator cyklu

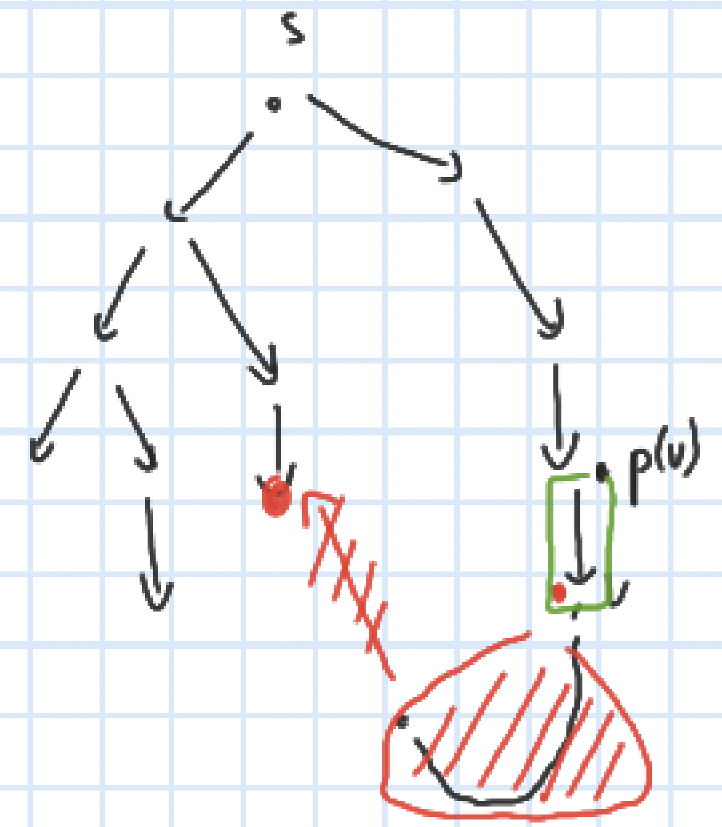
Przykład



Jeśli $d(v) = low(v)$ to krawędź $\{p(v), v\}$ jest mostem

takiej krawędzi do p(v) lub wierzchołka o niższym czasie odwiedzenia nie ma





Jeśli $low(v) < d(v)$ to $\{p(v), v\}$ nie jest mostem