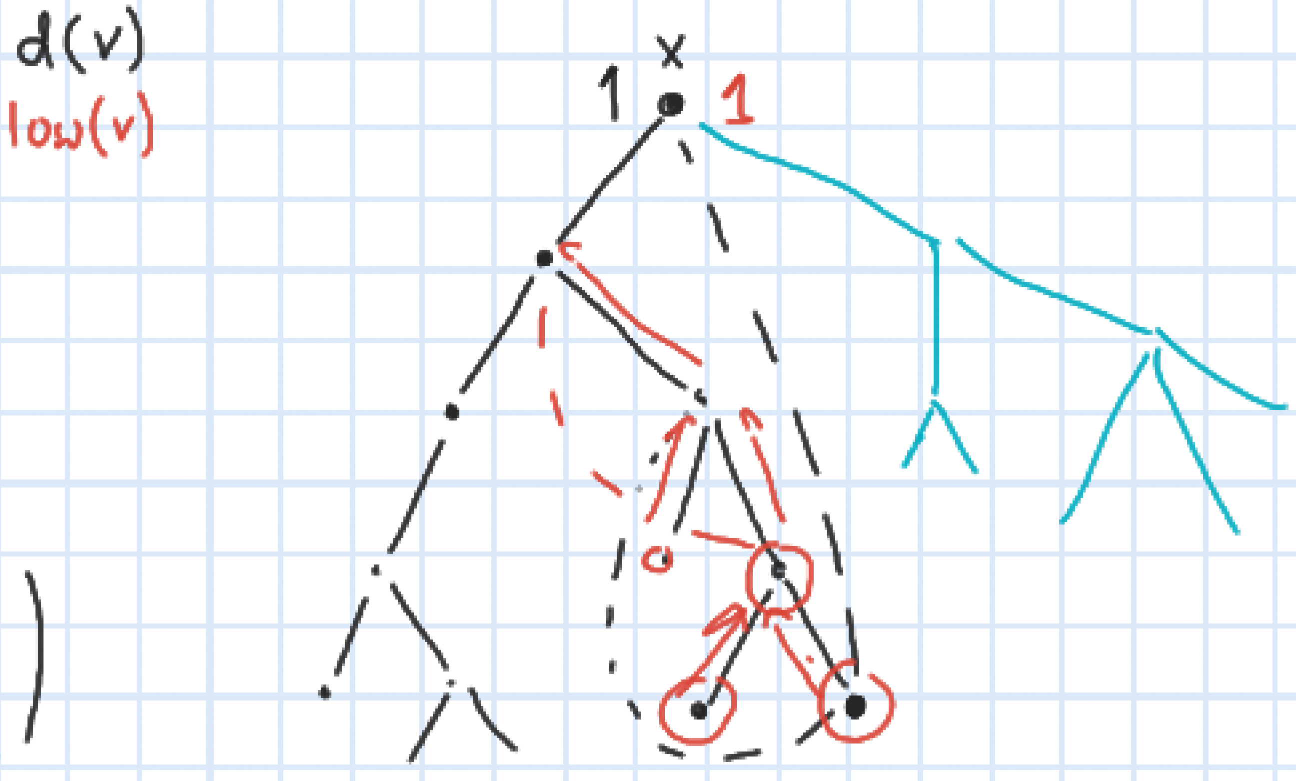
Algorytmy i Struktury Danych

Wykład 7

Most w grafie – krawędź, której usunięcie rozspójnia graf

Krawędź e jest mostem wtw. gdy nie leży na żadnym cyklu prostym



Algorytm

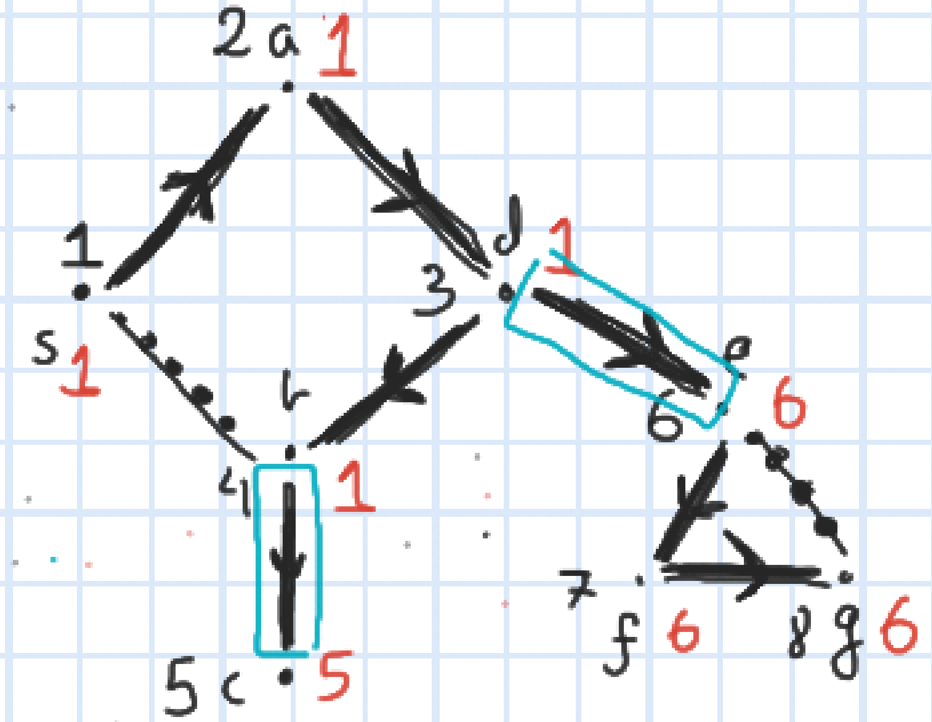
① DFS z zapisywaniem czasu odwiedzenia ($d(v)$ – czas odw. v)

② Dla każdego v obliczamy:

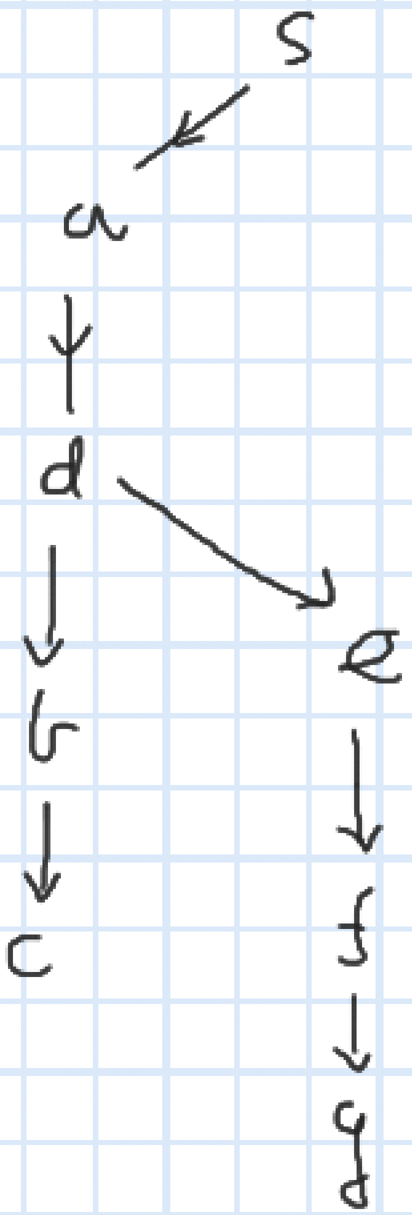
$$low(v) = \min\Big(d(v), \underbrace{\min d(u)}_{\text{istnieje krawędź wsteczna z } v \text{ do } u}, \underbrace{\min low(w)}_{w \text{ jest dzieckiem } v \text{ w drzewie DFS}} \Big)$$

③ Mosty to krawędzie $\{v, parent(v)\}$

gdzie $d(v) = low(v)$

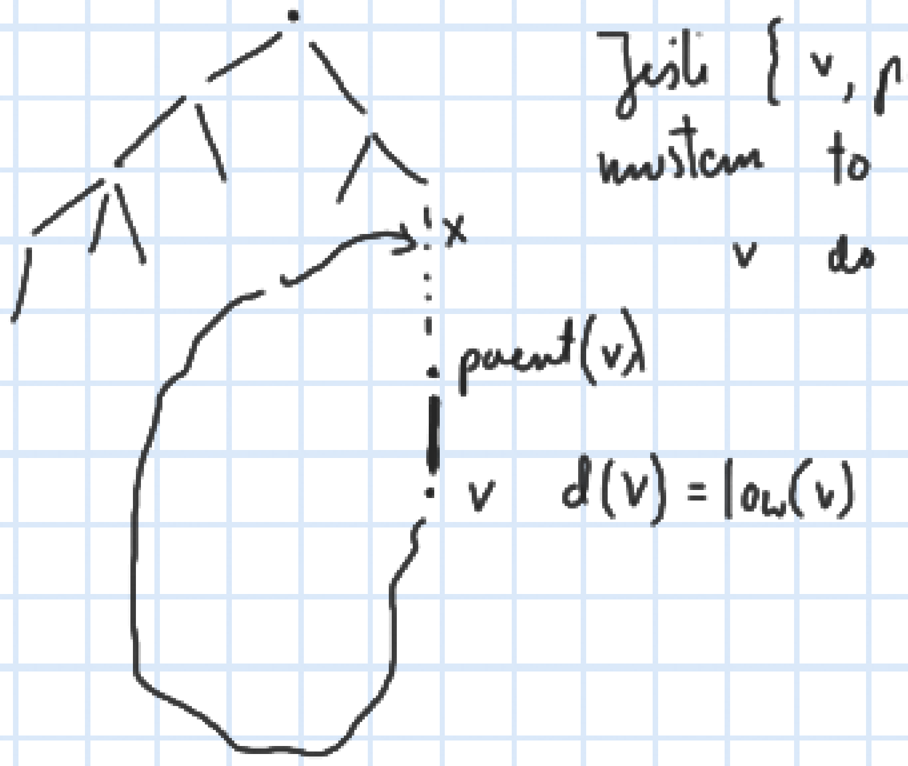


$d(v)$, $low(v)$



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

Dowód (nie wprost)

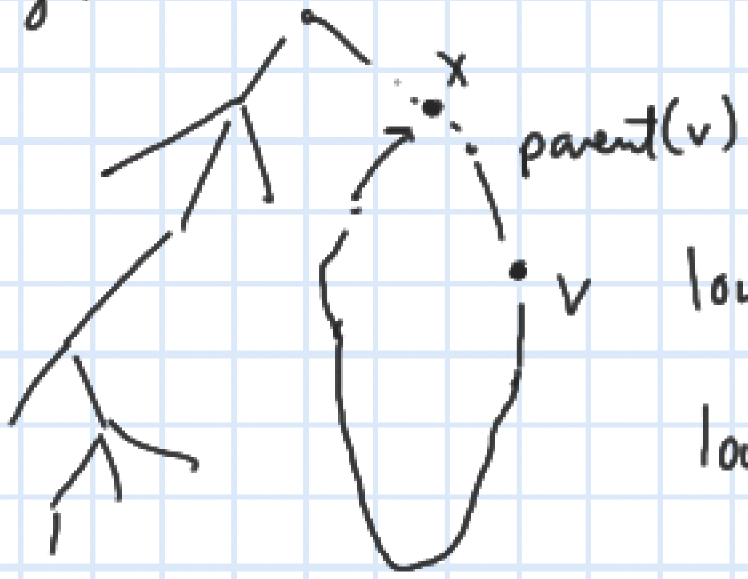


Jeśli $\{v, parent(v)\}$ nie jest mostem to istnieje ścieżka z v do $parent(v)$ nie używająca tej krawędzi — ta ścieżka musi zawierać krawędź wsteczną (do wierzchołka x, dla którego $d(x) < d(v)$)

$\Downarrow$

sprzeczność! ⟶ $low(v) \leq d(x) < d(v)$

Jeśli $d(v) \neq low(v)$ to $\{v, parent(v)\}$ nie jest mostem



$low(v) \neq d(v)$

$\Downarrow$

$low(v) < d(v)$ — to oznacza, że z v da się osiągnąć wierzchołek x, t.ż. $d(x) < d(v)$

Znajdowanie najkrótszych ścieżek w grafach ważonych

Reprezentacja

$G = (V, E)$ - graf

$w: E \to \mathbb{N}$ $(\mathbb{Z}, \mathbb{R})$ $\mathbb{Q}$

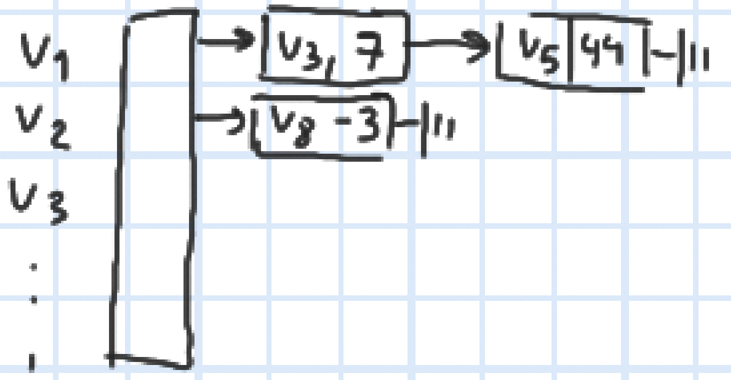
Reprezentacja macierzowa

W - macierz wag

$W[i][j]$ – waga krawędzi między v_i i v_j

∞ gdy nie ma krawędzi

Reprezentacja listowa



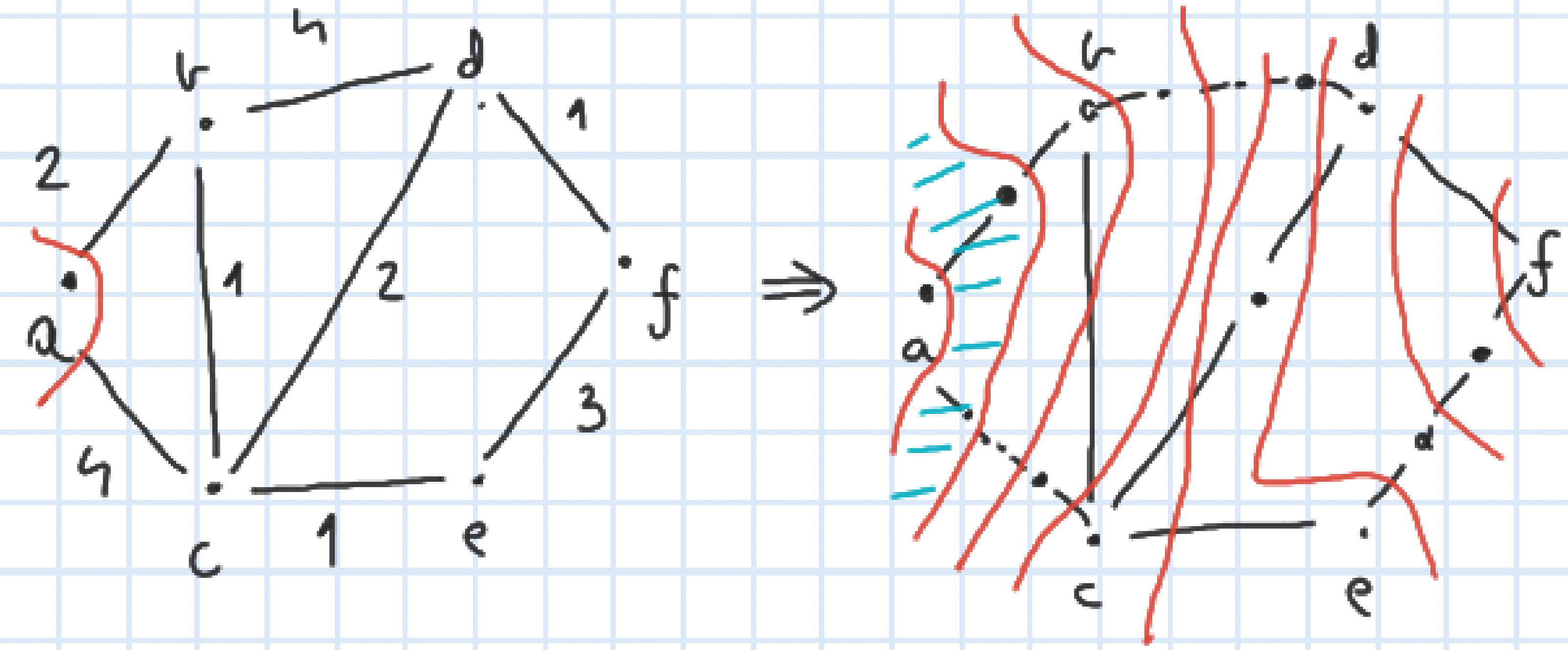
Warianty problemu najkrótszych ścieżek

- 1-1 – na elementarnym poziomie trudne do wykorzystania
- 1-wszyscy
- wszyscy-wszyscy

standardowe wersje problemu

Podejście elementarne (BFS z mnożeniem wierzchołków)

długości/wagi krawędzi to małe liczby naturalne



tworzenie sztucznych wierzchołków – nie tworzymy ich fizycznie, tylko w kolejce BFS przechowujemy pary (wierzchołek, licznik)

Algorytm Dijkstry – algorytm elementarny,

ale w każdym kroku od razu skaczemy do najbliższego prawdziwego wierzchołka

Nie wymaga wag naturalnych, ale wymaga wag nieujemnych

Notacja

$G = (V, E)$

$w(u, v)$ – odległość z u do v

$u.d$ – oszacowanie odległości ze źródła do d

$u.parent$ – poprzednik u na najkrótszej ścieżce

Złożoność obliczeniowa

$O(E \log V)$

Algorytm (start z $s \in V$)

źródło

1. Umieść wszystkie wierzchołki w kolejce priorytetowej z oszacowaniem odległości równym ∞

typu minimum

w praktyce często tylko ustawiamy pola $u.d$

2. Ustawiamy $s.d = 0$
3. Póki kolejka nie jest pusta:
   - wyjmij z kolejki u o minimalnej wartości $u.d$
   - dla każdej krawędzi $\{u, v\}$ (u, v) dla grafu skierowanego wykonaj relaksację

```
def relax(u, v):
    if v.d > u.d + w(u, v):
        v.d = u.d + w(u, v)
        v.parent = u
```

wymaga aktualizacji kolejki !

Príklad