Algorytmy i Struktury Danych

Wykład 8

Reprezentacja grafów ważonych

$G = (V, E)$

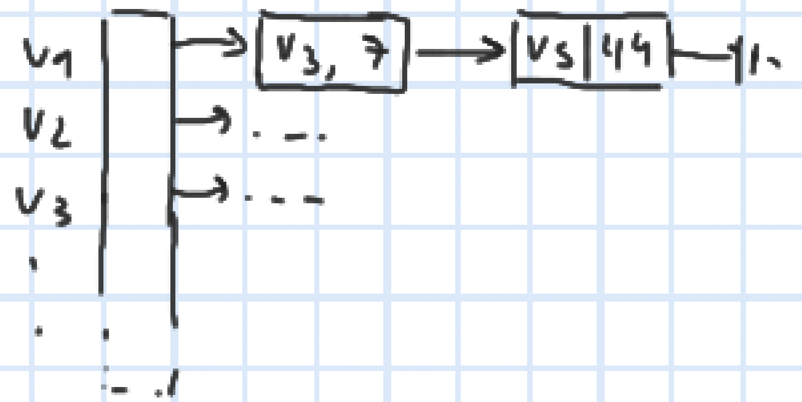
$w: E \to \mathbb{N} \quad (\mathbb{Z}, \mathbb{R}) \to \mathbb{Q}$

Reprezentacja macierzowa

W - macierz wag

$W[i][j]$ - waga krawędzi między wierzchołkami v_i oraz v_j (∞ jak nie ma)

Reprezentacja listowa

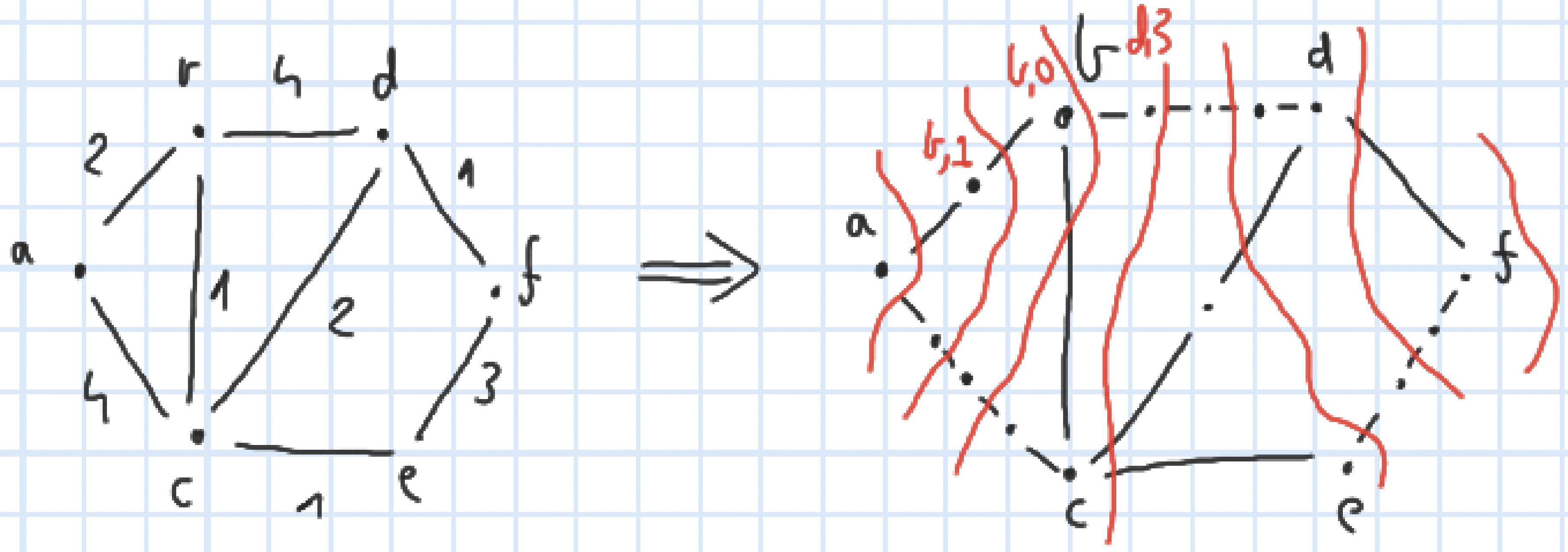


Problem znajdowania najkrótszych ścieżek

- 1 - 1 } na elementarnym poziomie trudne do wykorzystania
- 1 - wszystkie } standardowe wersje problemu
- wszystkie - wszystkie

Podejście elementarne / BFS

długości/wagi krawędzi to małe liczby naturalne



tworzymy sztuczne wierzchołki odpowiednio dzieląc krawędzie

Algorytm Dijkstry – algorytm elementarny, ale w każdym kroku skacze do najbliższego prawdziwego wierzchołka

nie wymagamy wag naturalnych, ale mogą być nieujemne

Notacja

$G = (V, E)$

$w(u,v)$ – odległość z u do v

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

$u.parent$ – poprzednik na najkrótszej ścieżce ze źródła

$O(E \log V)$ – złożoność algorytmu Dijkstry dla rep. listowej z kopcem binarnym

$O(V^2)$ – rep. macierzowa z "liniową kolejką"

Algorytm (start z $s \in V$)

1. Umieść wszystkie wierzchołki w kolejce priorytetowej z oszacowaniem odległości ∞ } w praktyce realizowane inaczej

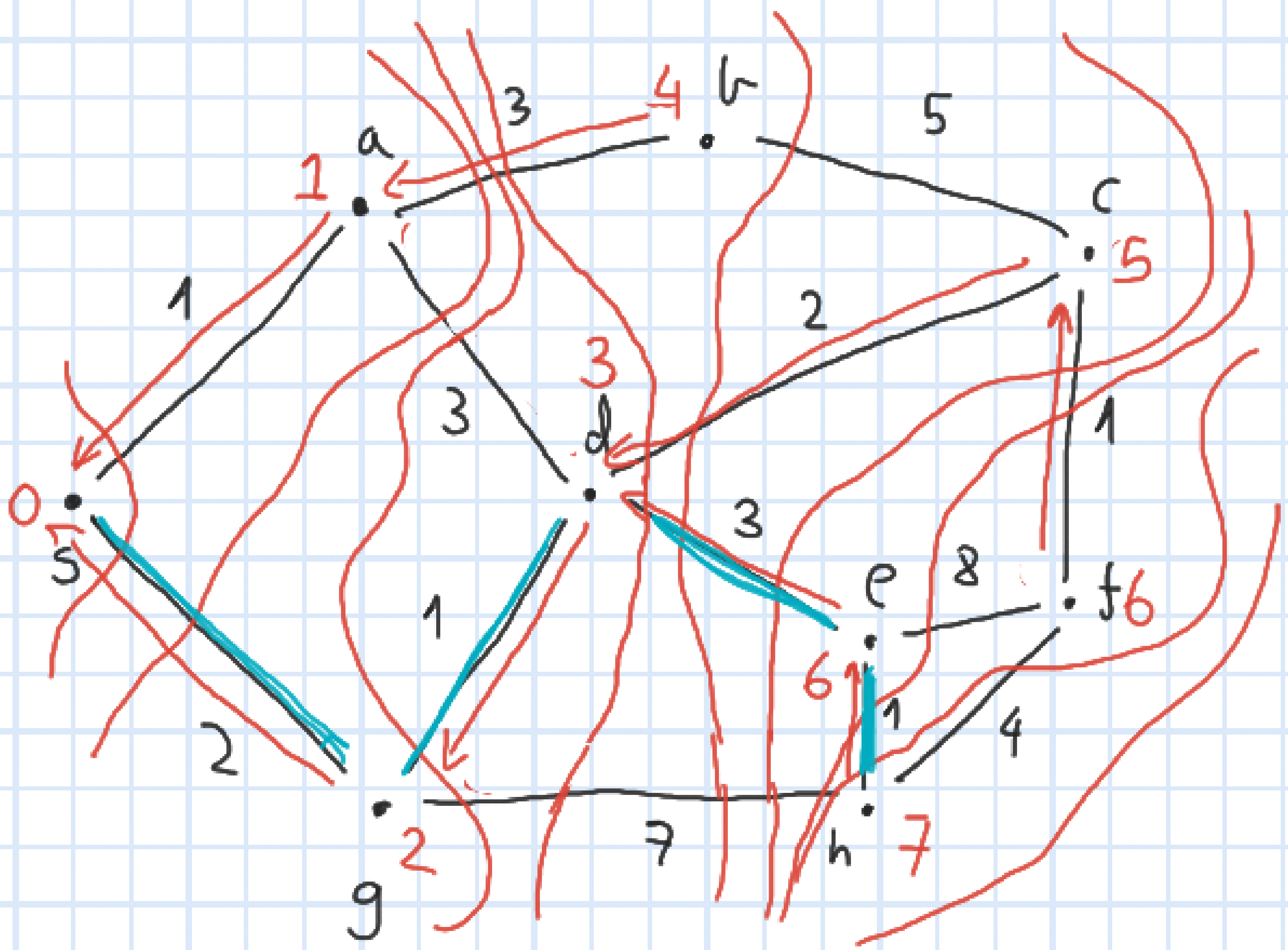
2. Zmień odległość s na 0

3. Póki wierzchołki są w kolejce
   - wyjmij z kolejki wierzchołek u o minimalnej wartości $u.d$
   - dla każdej krawędzi $\{u,v\}$ wykonaj relaksację

(u,v) – dla grafu skierowanego

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

Przykład



Dlaczego algorytm Dijkstry jest poprawny?

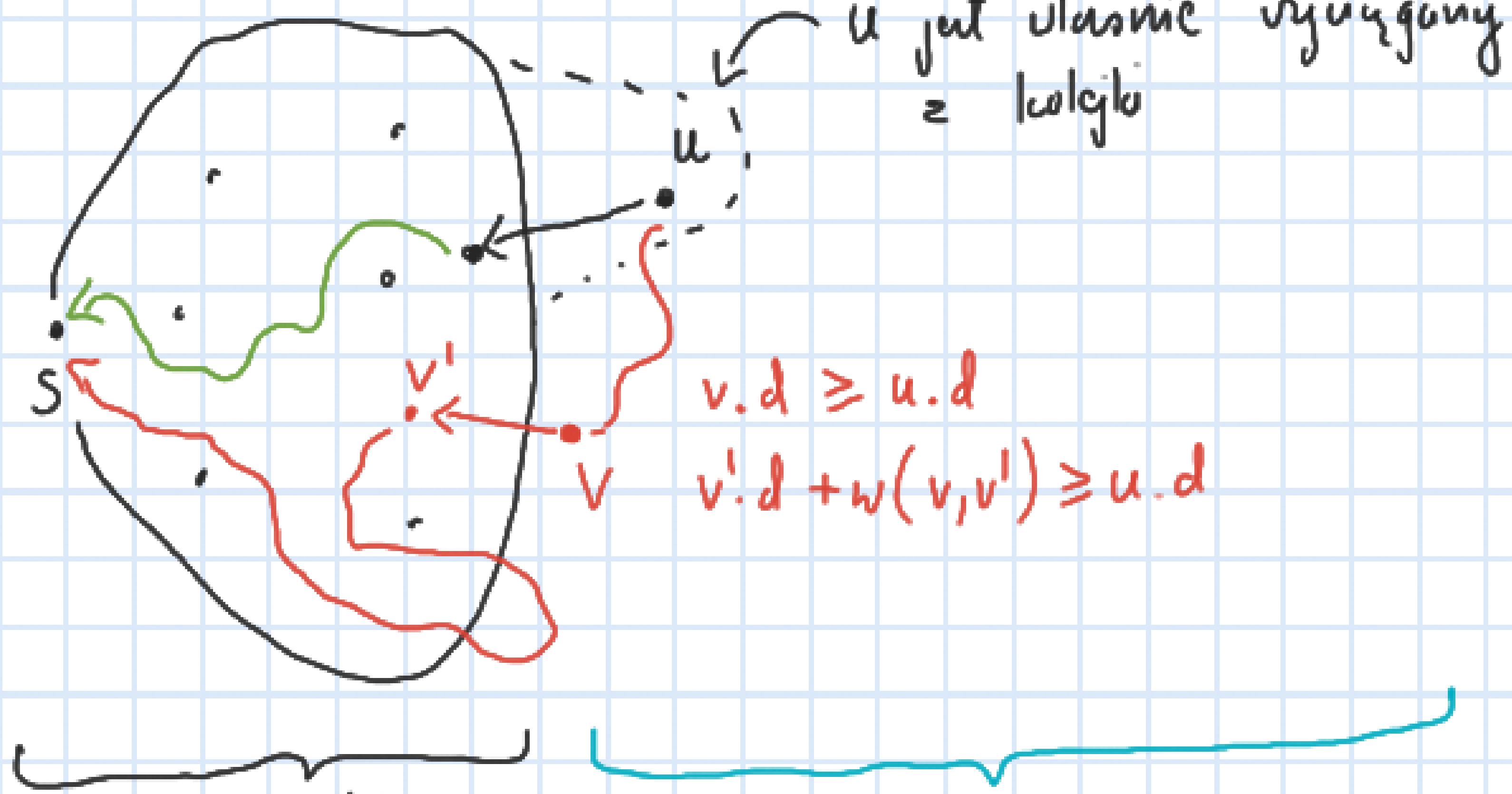
- bo realizuje BFS z dodanymi wierzchołkami
  (nieprawidłowy argument jeśli wagi nie są naturalne)
- dowód indukcyjny

<u>tw</u> Gdy algorytm Dijkstry wyciąga wierzchołek u z kolejki, to jego pole u.d zawiera długość najkrótszej ścieżki z s do u

<u>Dowód</u> (przez indukcję)

Podstawa indukcji — dla s jest to w oczywisty sposób prawda

<u>Krok indukcyjny</u>



wierzchołki, które zostały wyjęte z kolejki

ich pola d mają prawidłowe wartości (tak samo pola parent)

wierzchołki w kolejce

Algorytm Bellmana-Forda

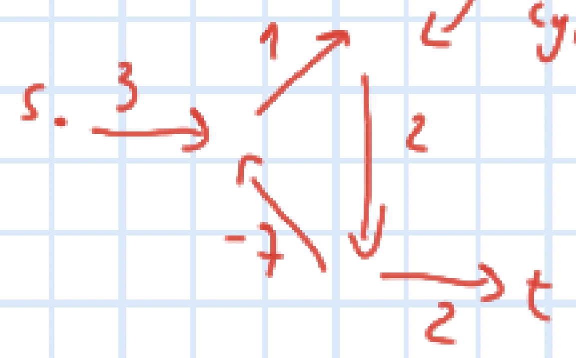
Najkrótsze ścieżki gdy dopuszczamy ujemne wagi

z def. graf musi być skierowany

① Inicjalizacja

```
for v ∈ V:
    v.d = ∞
    v.parent = None
s.d = 0
```

musimy wykrywać takie cykle



② Relaksacja

```
for i in range(|V| - 1):
    for (u,v) ∈ E:
        Relax(u,v)
```

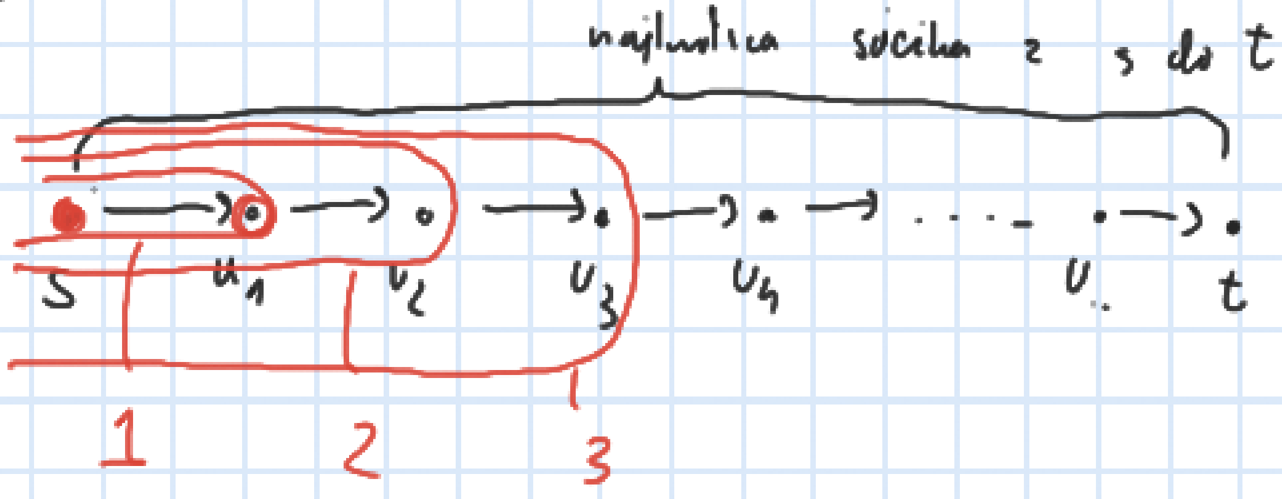
$$\boxed{O(VE)}$$

③ Weryfikacja

czy dla każdego $(u,v) \in E$:

$v.d \le u.d + w(u,v)$?

Czemu kroki ① + ② dają dobry wynik, jeśli nie ma cykli o ujemnej wadze?

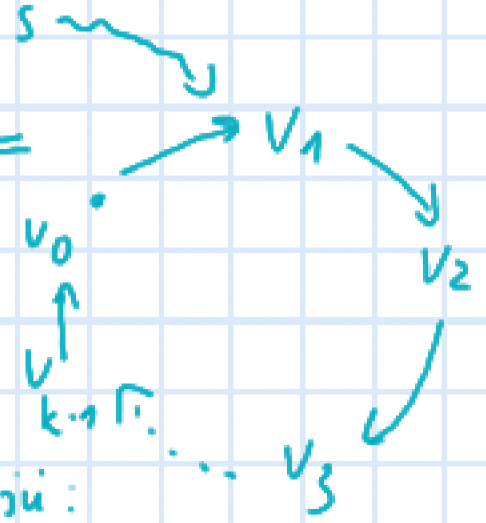


Każda iteracja relaksuje co najmniej jedną dalej krawędź najkrótszej ścieżki

Czemu ③ wykrywa ujemne cykle

Jeśli weryfikacja tego nie wykryła to dla każdego i:

$v_i.d \le v_{i-1}.d + w(v_{i-1}, v_i)$



$$\sum_{i=0}^{k-1} w(v_i, v_{i+1}) < 0$$

$$\{ v_k = v_0$$

Po zsumowaniu tych nierówności:

$$\underbrace{\boxed{\sum_{i=1}^{k} v_i.d}}_{x} \le \underbrace{\boxed{\sum_{i=1}^{k}\Big(v_{i-1}.d}}_{x} + \underbrace{\boxed{w(v_{i-1}, v_i)\Big)}}_{<0}$$

sprzeczność

Najkrótsze ścieżki między każdą parą wierzchołków

- $|V|$ wywołań Dijkstry $O(VE \log V)$
- $|V|$ wywołań Bellmana-Forda $O(V^2 E)$

Istnieje specjalizowany algorytm Floyda-Warshalla który znajduje rozwiązanie w czasie $O(V^3)$