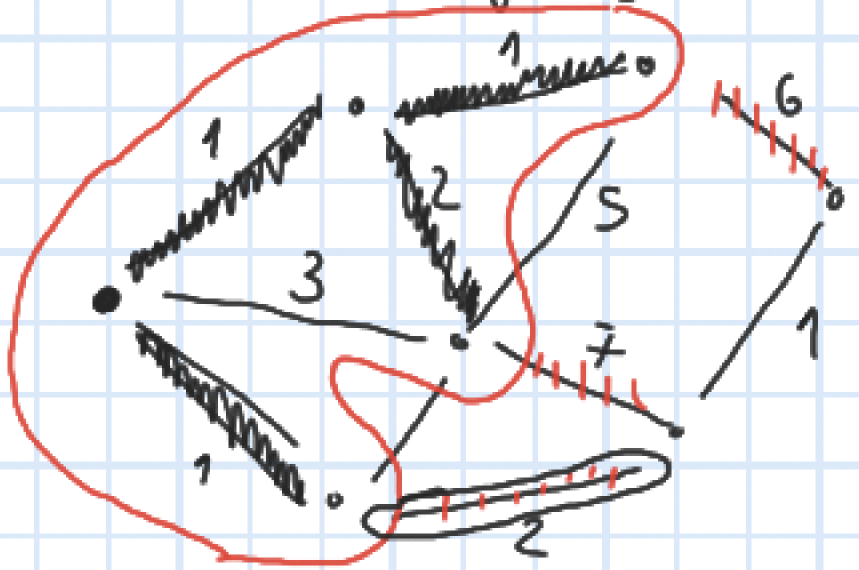
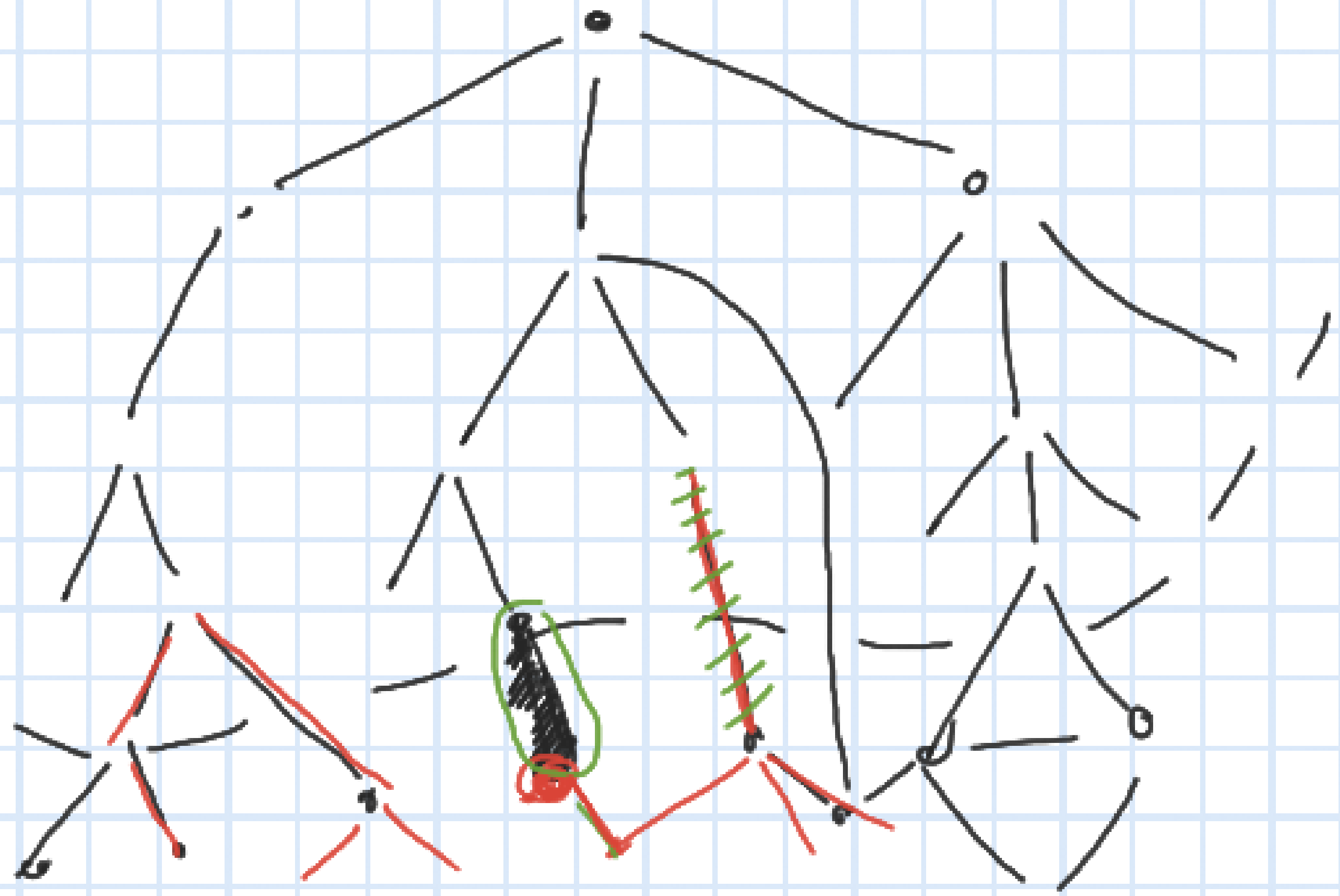
ASD - Wykład 12

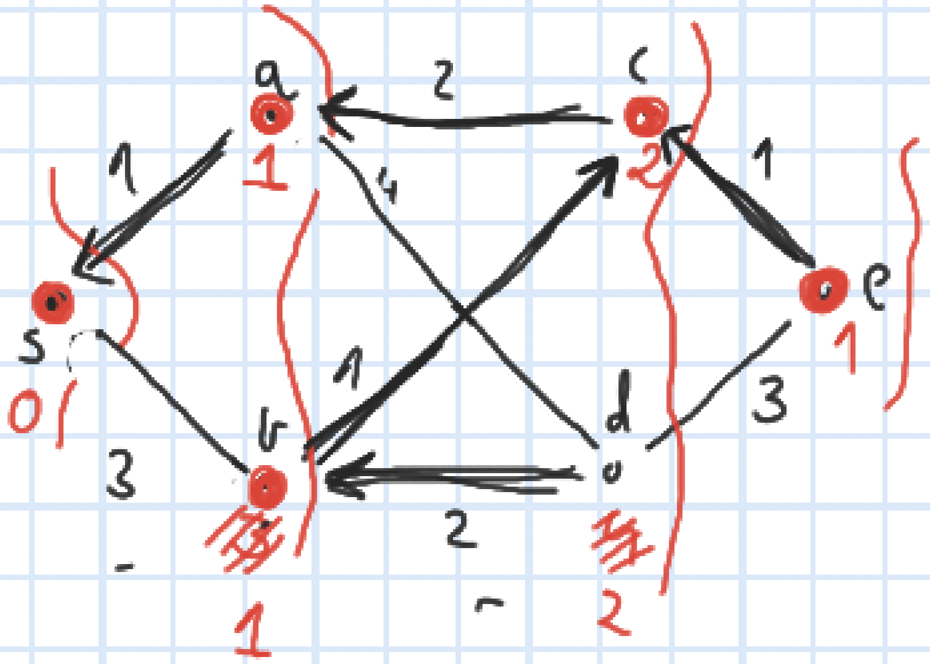
Algorytm Prima znajdowania minimalnego drzewa rozpinającego → Opis algorytmu

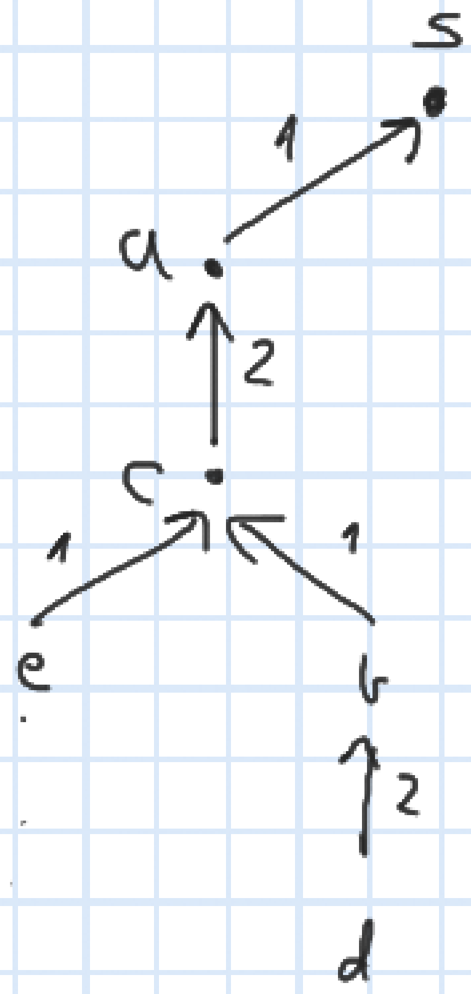




1. Startujemy z wierzchołka s
2. Wszystkie wierzchołki umieszczamy w kolejce priorytetowej z wagą ∞
3. Zamieniamy wagę s na 0
4. Póki są wierzchołki w kolejce:
   - wyjmij wierzchołek t o minimalnej wadze
   - dla każdej krawędzi $\{t,u\}$, jeśli waga $u \geq w(t,u)$ to zamień wagę u na $w(t,u)$ (i uaktualnij u.parent)

Przykład wykonania algorytmu Prima



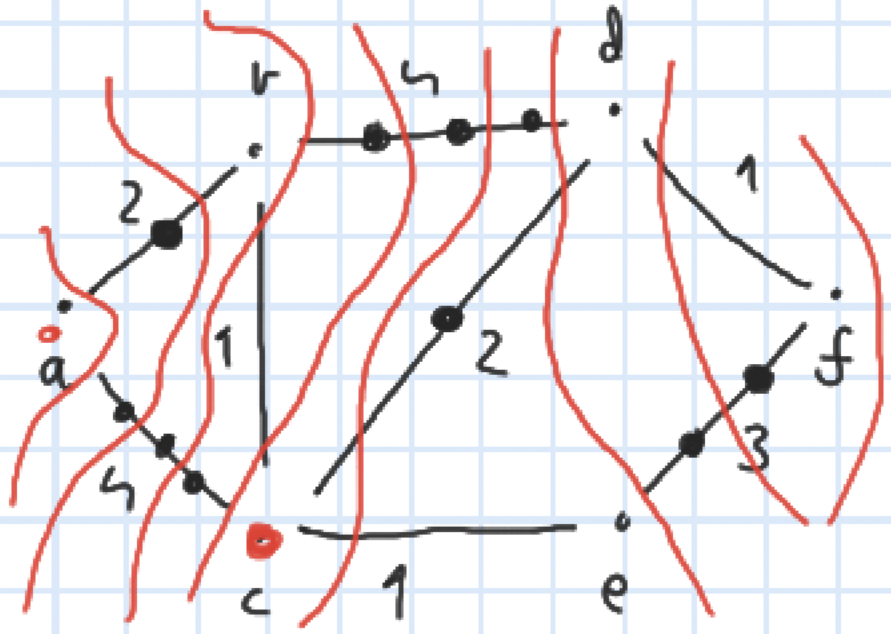


Problem znajdowania najkrótszych ścieżek

Warianty

- 1 - 1 } na elementarnym poziomie trudne do wykazania
- 1 - wszyscy
- wszyscy - wszyscy } standardowe wersje problemu

Podejście elementarne



BFS na sztucznie rozmnożonych krawędziach

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

Wagi nie muszą być naturalne, ale muszą być nieujemne

Notacja

$G = (V, E)$, $w: E \to \mathbb{R}_+$ – wejściowy graf ważony

$w(u,v)$ – waga krawędzi $\{u,v\}$

$u.d$ – oszacowanie odległości z wierzchołka startowego

$u.parent$ – poprzednik na najkrótszej ścieżce (ze startowego do u)

Algorytm (startujemy z wierzchołka s)

złożoność $O(E \log V)$

① Umieść wszystkie wierzchołki w kolejce priorytetowej (min) z oszacowaniem odległości ∞

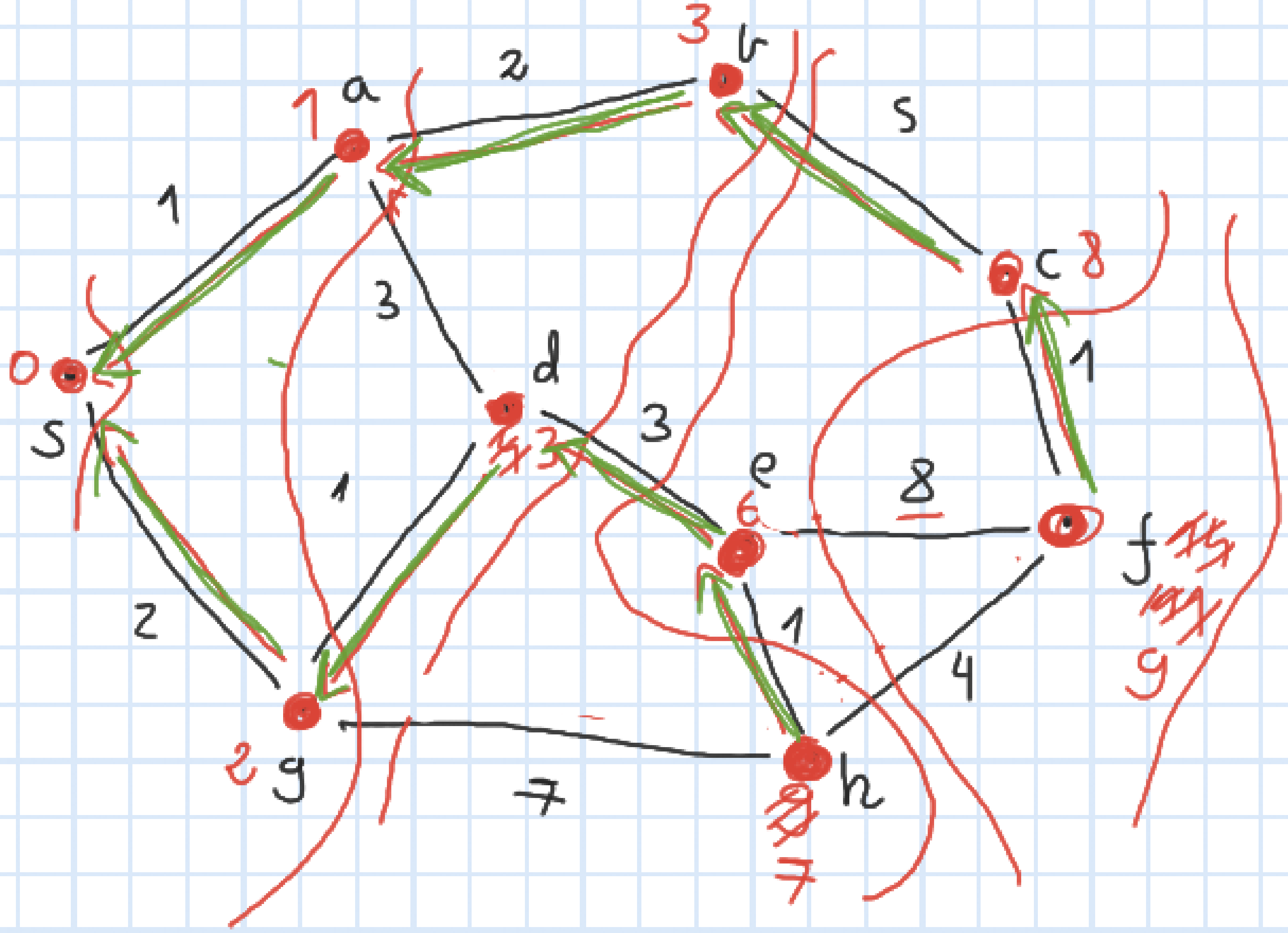
② zmień odległość s na 0

③ póki są wierzchołki w kolejce:

- wyjmij z kolejki wierzchołek u (o minimalnej wartości $u.d$)
- dla każdej krawędzi $\{u,v\}$ wykonaj relaksację:

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

Przykład wykonania algorytmu Dijkstry

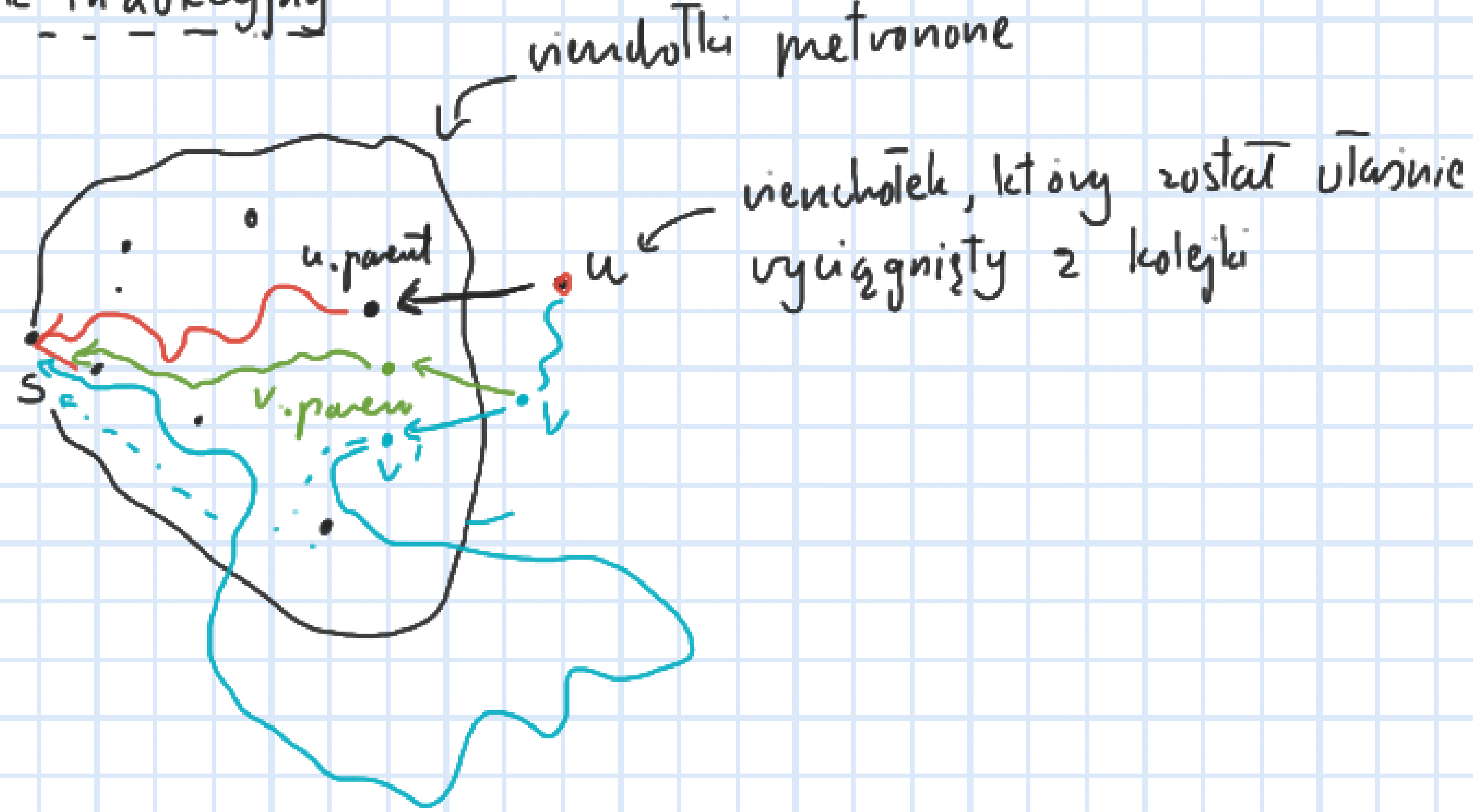


tw Gdy algorytm Dijkstry wyciąga wierzchołek u z kolejki, to jego pole u.d zawiera długość najkrótszej ścieżki z s do u

Dowód (przez indukcję)

dla wierzchołka startowego oczywiście zachodzi

krok indukcyjny



Algorytm Bellmana-Forda

– najkrótsze ścieżki jeśli wagi mogą być ujemne

① Inicjalizacja

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

② Relaksacje

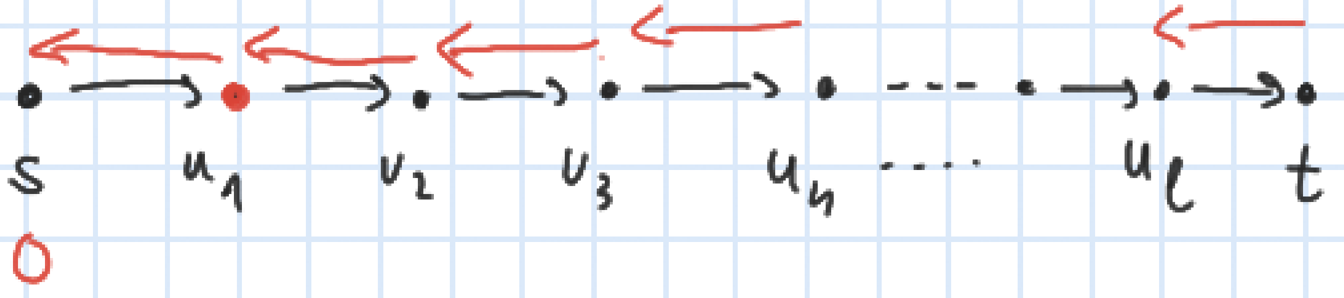
```
for i in range(|V|-1):
    for (u,v) ∈ E:
        Relax(u,v)
```

③ Weryfikacja

czy dla każdej $(u,v) \in E$:

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

Czemu to działa?



złożoność

$O(VE)$