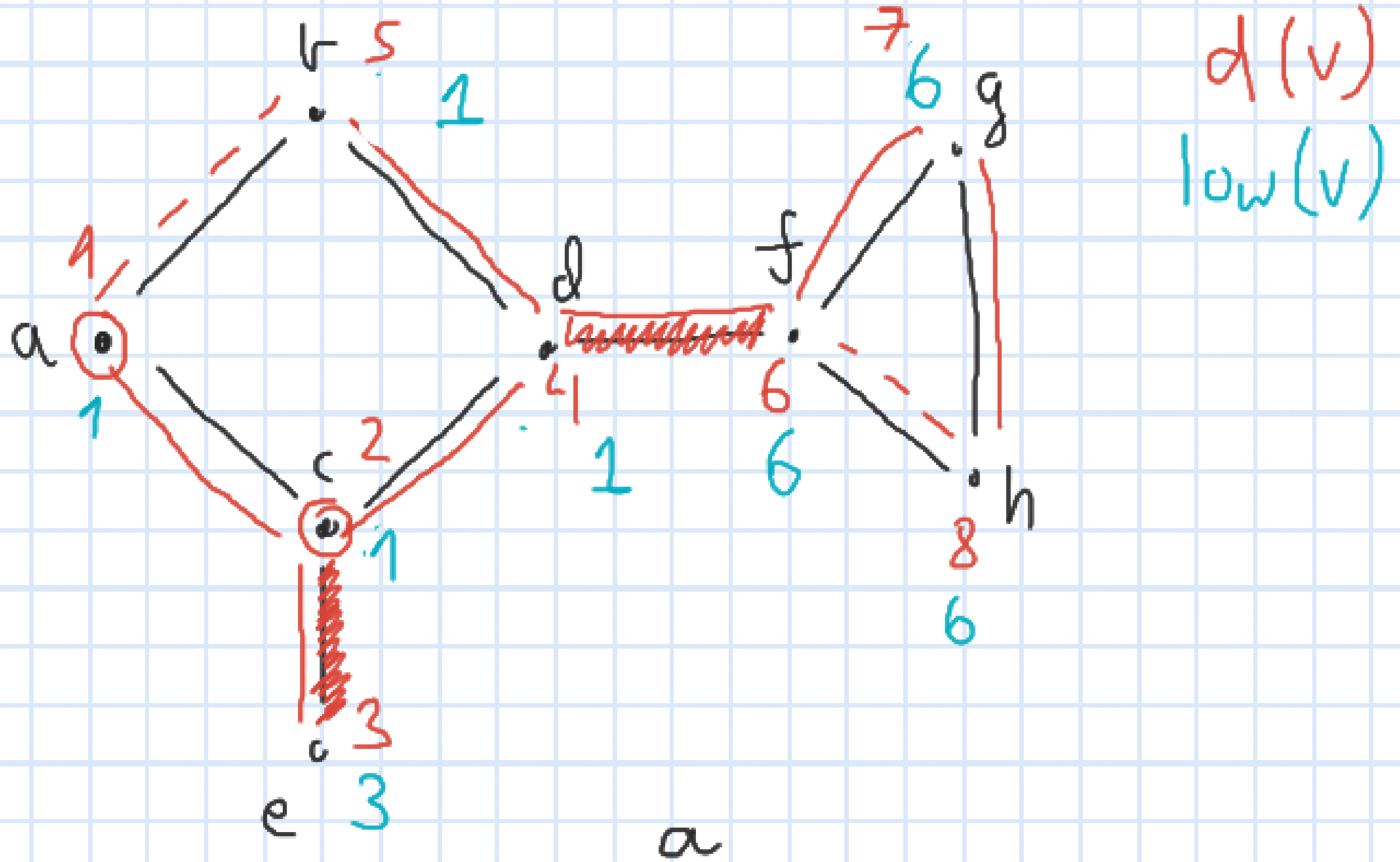
ASD - Wykład 11

<u>def</u> Krawędź e w spójnym grafie nieskierowanym jest mostem jeśli jej usunięcie rozspójnia graf

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

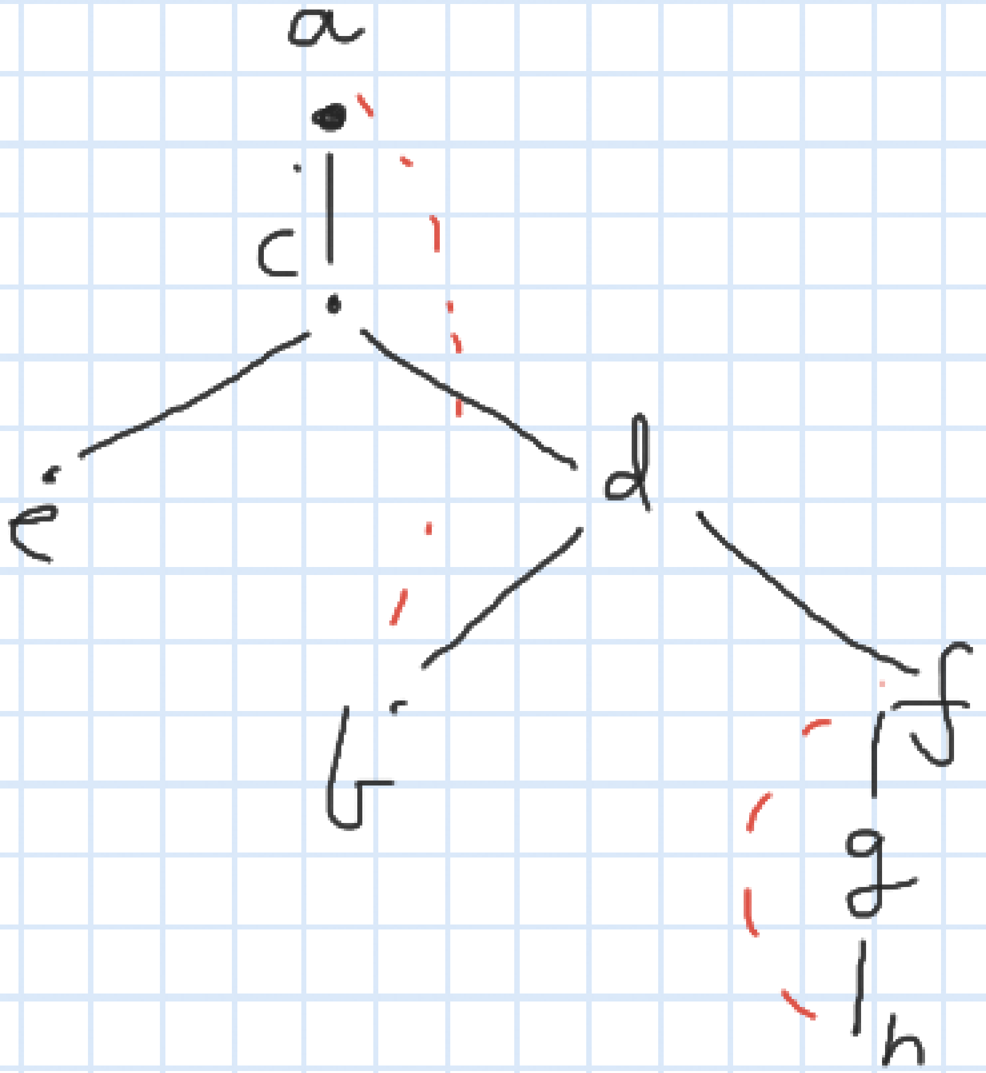


Algorytm

① Wykonaj DFS, dla każdego wierzchołka v zapamiętując czas odwiedzenia $d(v)$

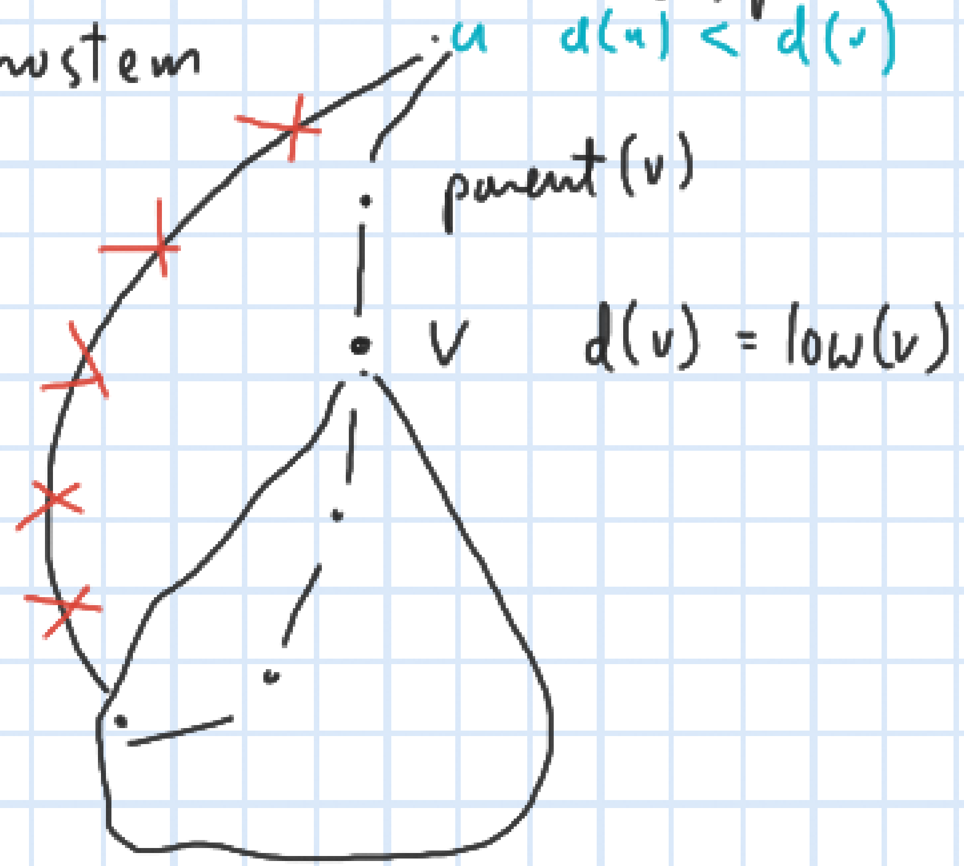
② Dla każdego wierzchołka v obliczamy

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

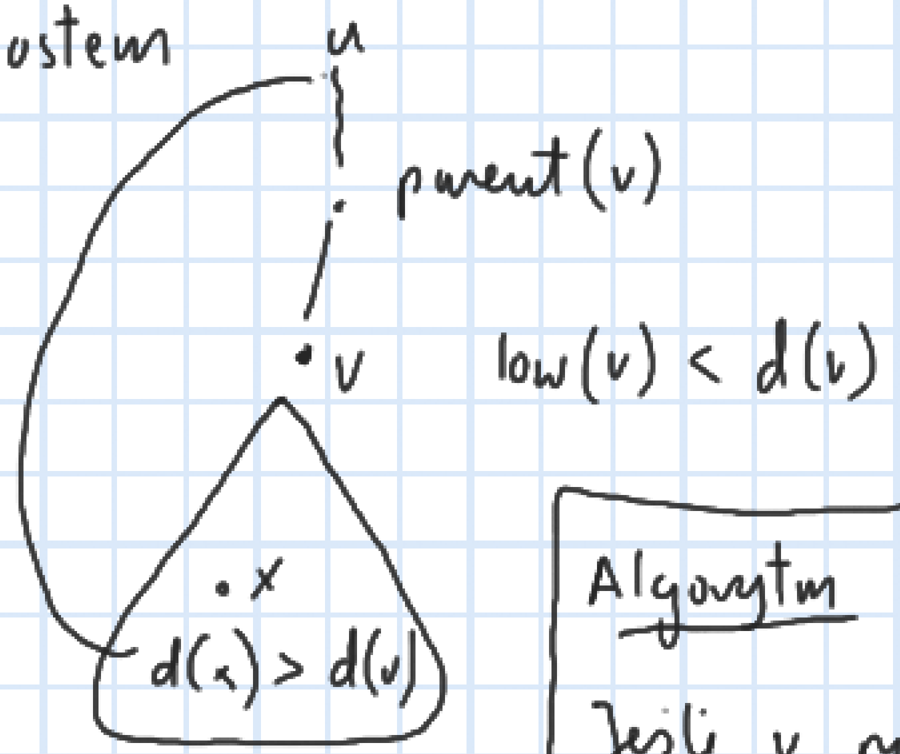


③ Mosty to krawędzie $\{v, parent(v)\}$ gdzie $d(v) = low(v)$

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

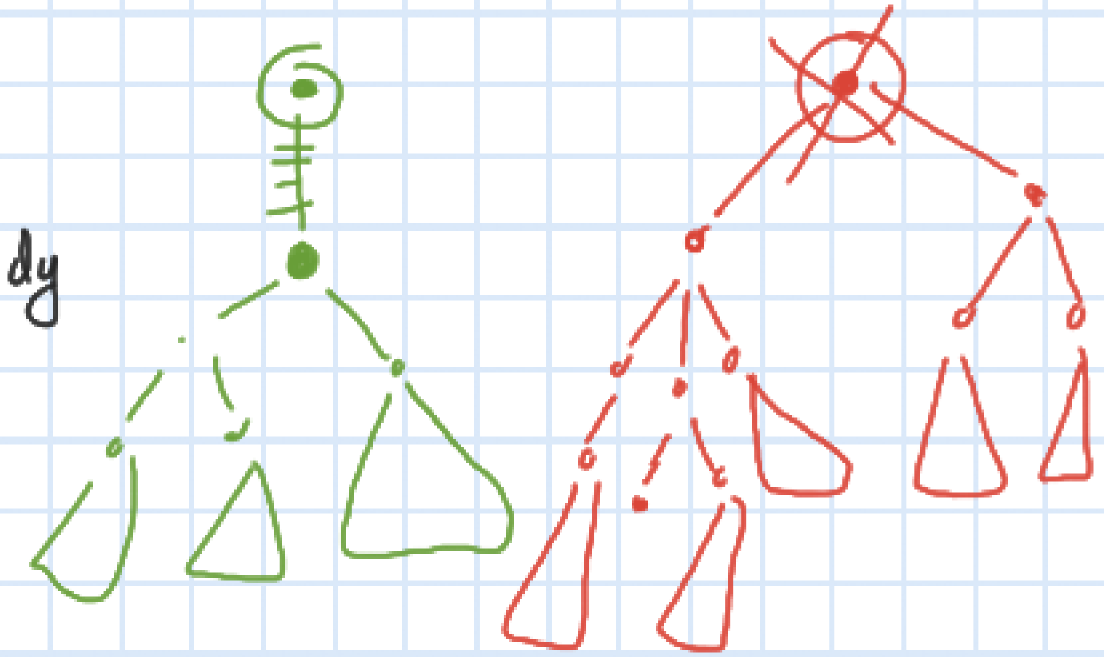


Jeśli $d(v) \neq low(v)$ to $\{v, parent(v)\}$ nie jest mostem

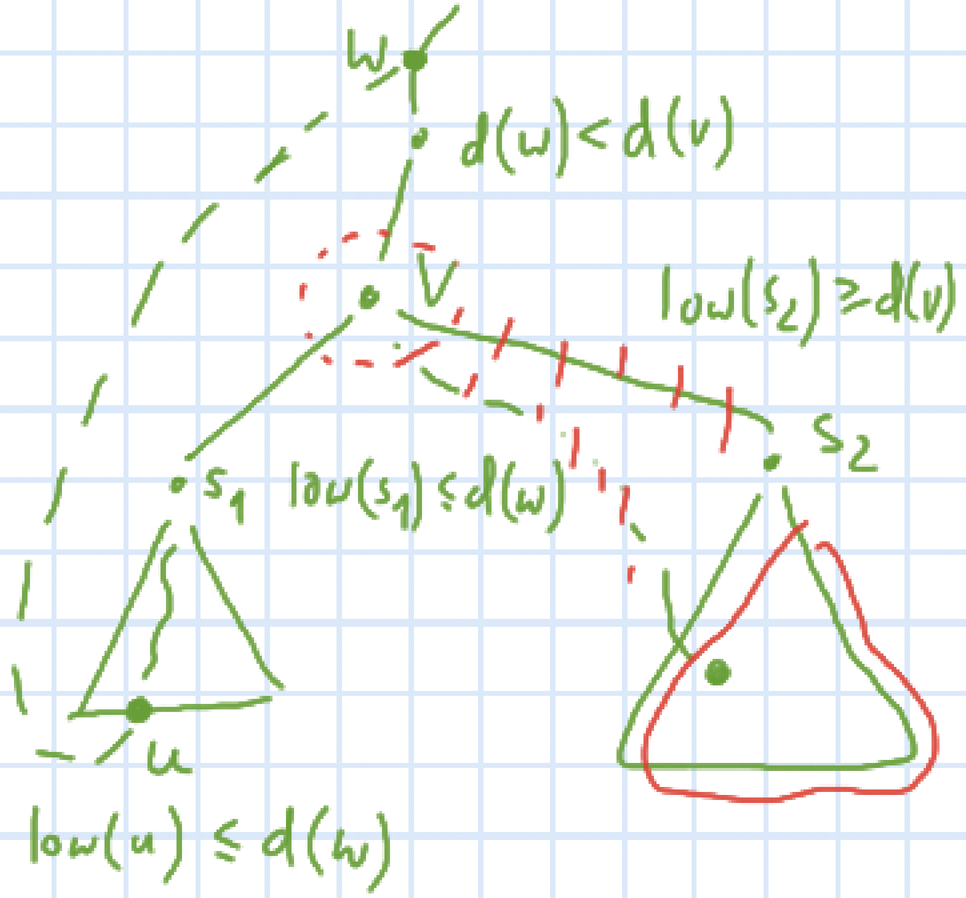


<u>def</u> Punkt artykulacji nieskierowanego grafu G to taki wierzchołek v, którego usunięcie zwiększa liczbę spójnych składowych

<u>tw</u>
Korzeń drzewa DFS jest punktem artykulacji wtw. gdy posiada co najmniej dwoje dzieci w drzewie DFS



<u>tw</u> Wierzchołek nie będący korzeniem jest punktem artykulacji wtw. gdy dla przynajmniej jednego syna nie istnieje krawędź wsteczna $\{u, w\}$ taka że u to potomek v a w jest przodkiem v



<u>Algorytm</u>

Jeśli v posiada syna s takiego, że $low(s) \geq d(v)$ to v jest punktem artykulacji (korzeń osobno)

Algorytmy na grafach ważonych

- minimalne drzewo rozpinające
  (MST - minimal spanning tree)
- najkrótsze ścieżki

Reprezentacja grafów ważonych

$G = (V, E)$

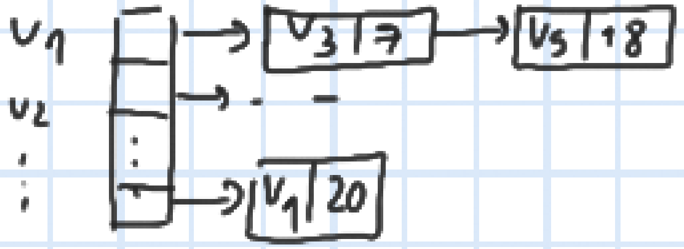
$w : E \to \mathbb{N} \quad (\mathbb{Z}, \mathbb{Q}, \mathbb{Q}_+)$

Reprezentacja macierzowa

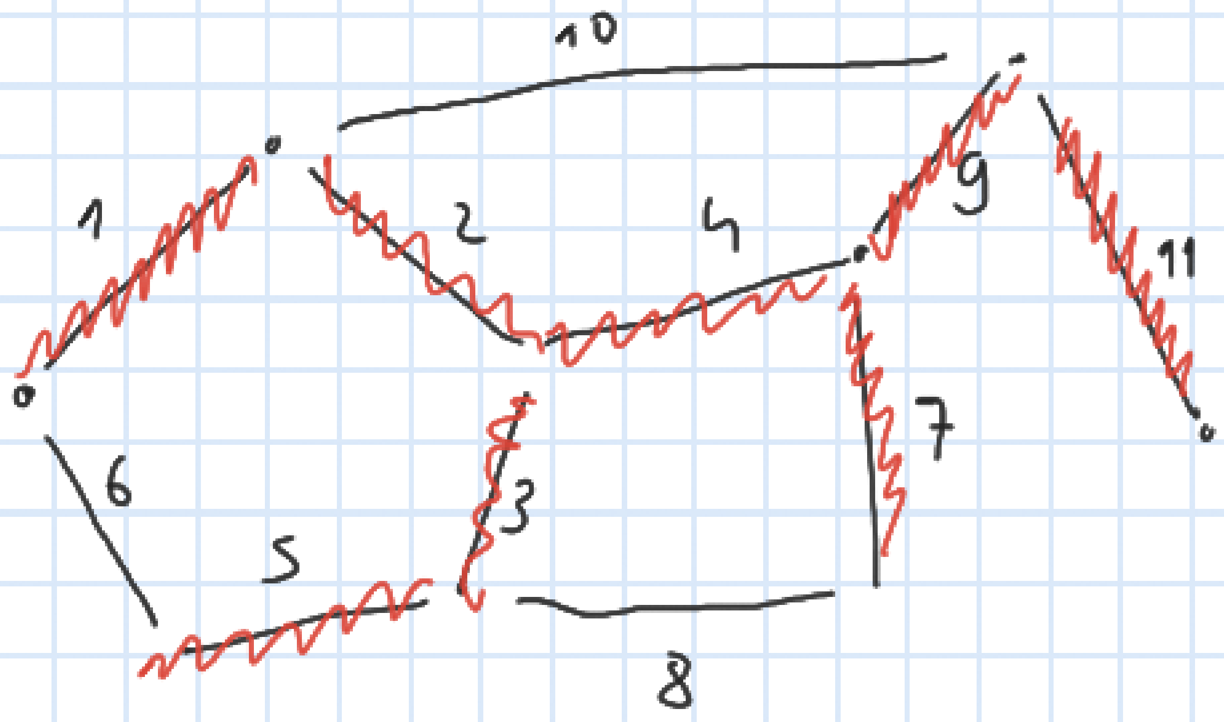
W - macierz wag

$W[i][j]$ - waga krawędzi między wierzchołkami v_i oraz v_j

Reprezentacja listowa



Minimalne drzewo rozpinające



MST to zbiór krawędzi, które łączą wszystkie wszystkie wierzchołki (w spójny podgraf) i których suma wag jest minimalna

Obserwacja

$G = (V, E)$, $w : E \to \mathbb{N}$ (I+)

Jeśli $A \subseteq E$ jest podzbiorem krawędzi pewnego MST dla G oraz $e = \{u, v\}$ jest krawędzią, taką że:

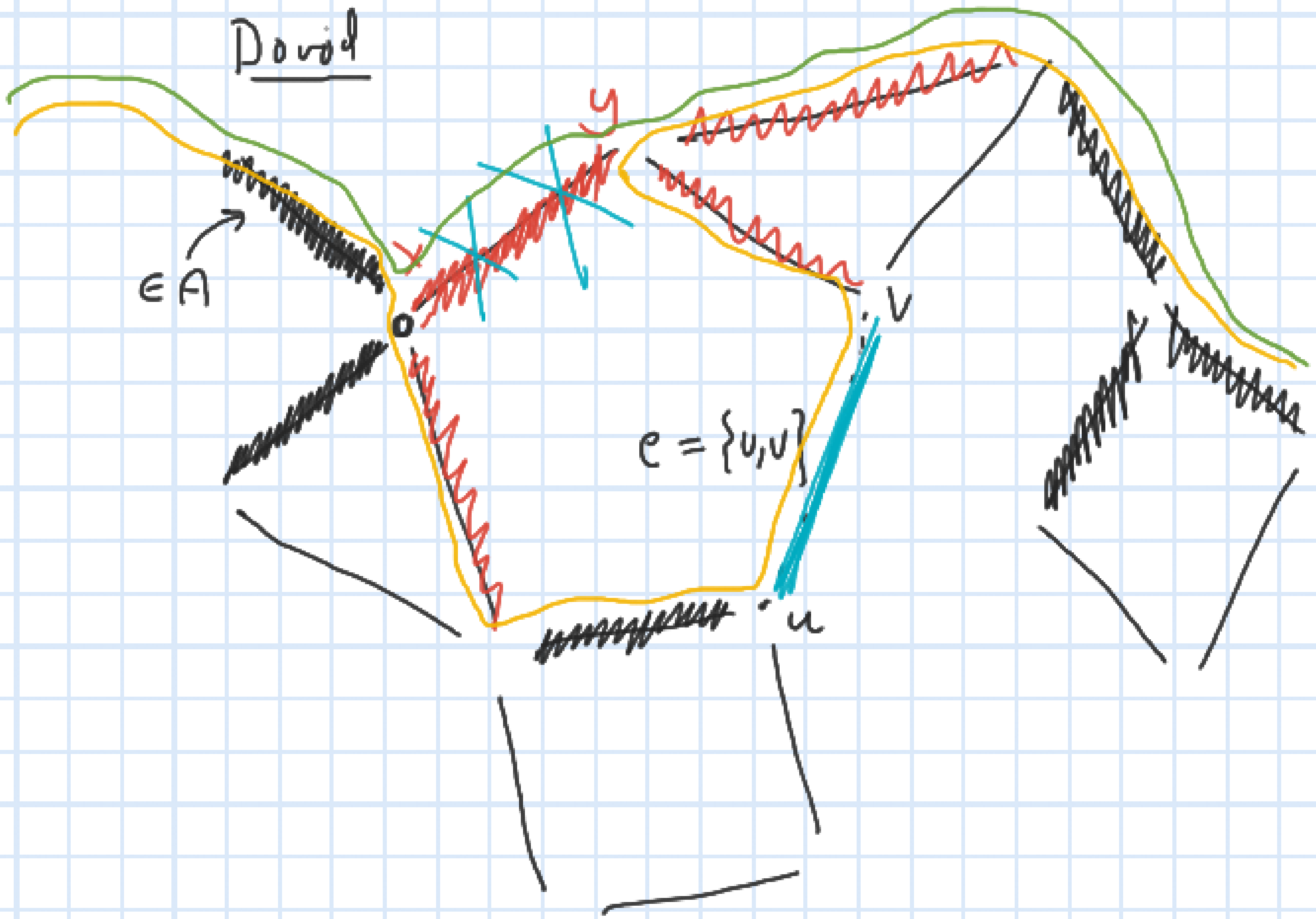
a) $e \notin A$

b) $A \cup \{e\}$ nie tworzy cyklu

c) krawędź e ma minimalną wagę spośród krawędzi spełniających a) i b)

to $A \cup \{e\}$ jest podzbiorem krawędzi pewnego MST dla G

Dowód



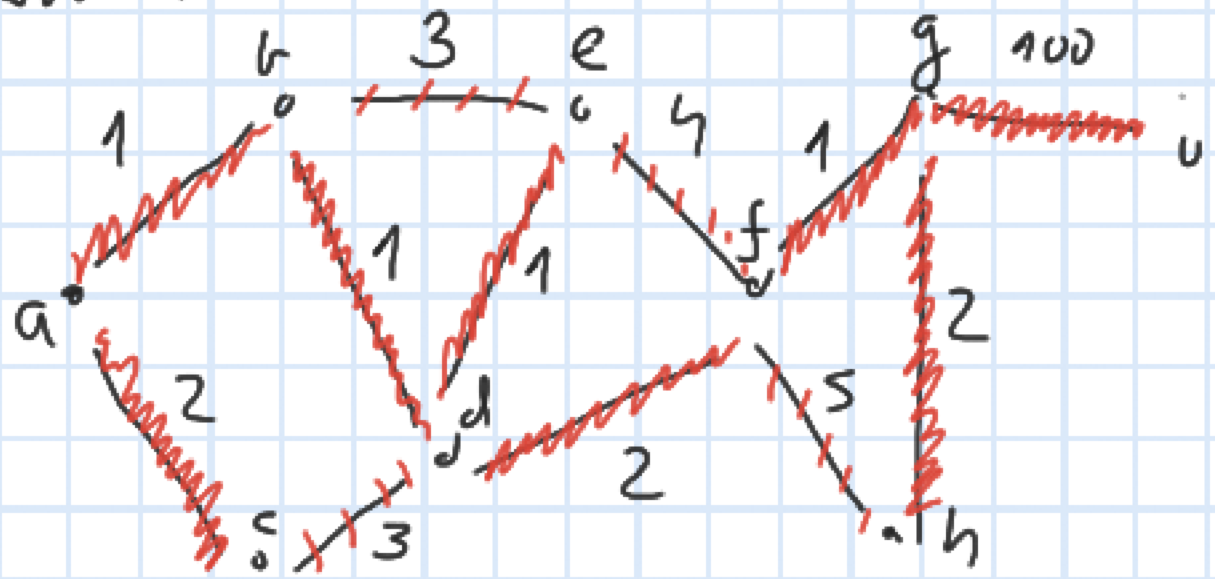
Jeśli $e = \{u, v\}$ nie należy do żadnego MST zawierającego A to jakaś inna krawędź $\{x, u\} \notin A$ służy do zapewnienia trasy z u do v

Algorytm Kruskala znajdowania MST

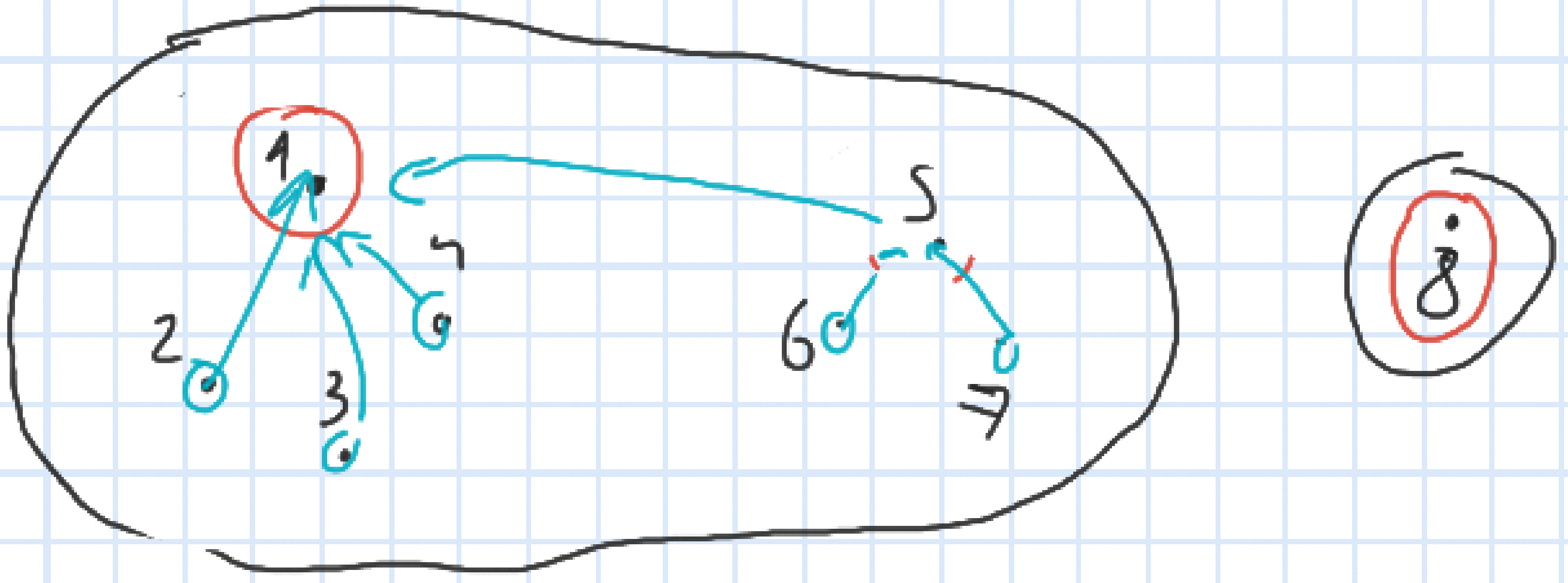
1. Posortuj krawędzie po wagach
2. $A = \emptyset$
3. Przeglądaj krawędzie e w kolejności niemalejących wag

   Jeśli $A \cup \{e\}$ nie zawiera cyklu to $A := A \cup \{e\}$

4. Zwróć A



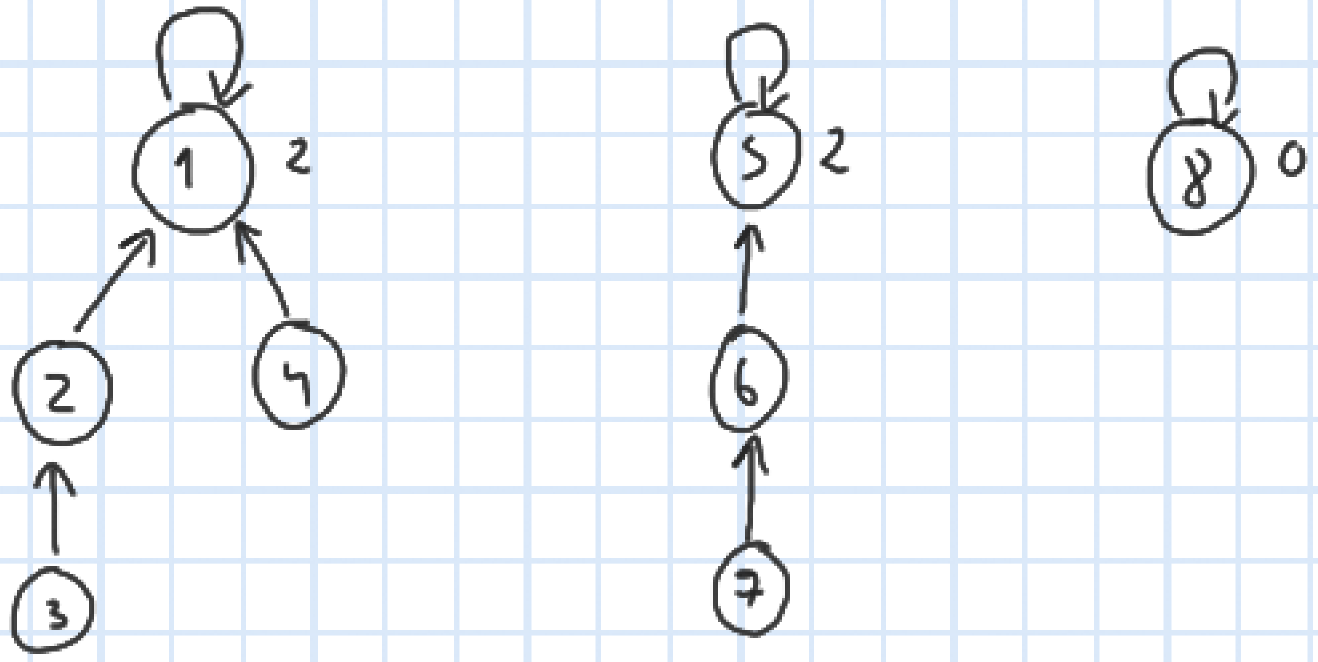
Lasy zbiorów rozłącznych / Find-Union



Trzy operacje

- makeset – stwórz nowy zbiór jednoelementowy
- find – znajdź reprezentanta zbioru do którego należy dany element
- union – połącz dwa zbiory

Przykład reprezentacji zbiorów rozłącznych




```
class Node:
    def __init__(self, value):
        self.parent = self
        self.value = value
        self.rank = 0
```

```
def find(x):
    if x.parent ≠ x:
        x.parent = find(x.parent)
    return x.parent
```

```
def union(x, y):
    x = find(x)
    y = find(y)
    if x == y: return

    if x.rank > y.rank:
        y.parent = x
    else:
        x.parent = y
        if x.rank == y.rank:
            y.rank += 1
```

Obserwacja 1 Ranga to górne ograniczenie na wysokości drzewa

Obserwacja 2 Jeśli drzewo ma rangę n to zawiera co najmniej 2^n węzłów

Obserwacja 3 Jeśli mamy łącznie n elementów to ciąg m operacji find i union ma złożoność $O(m \log n)$

tw Ciąg m operacji find i union, gdy mamy łącznie n elementów, wynosi $O(m \log^* n)$