Grafy

Zapowiedź: *Word ladder* (*dublety*) – (jednoosobowa) gra polegająca na przeprowadzeniu wybranego *legalnego* słowa na inne sekwencją ruchów polegających na zamianie jednej litery w słowie. Słowa uzyskane w kolejnych krokach muszą być legalne.

Cel: zminimalizować liczbę potrzebnych ruchów.
Przykłady:
MATH → MATS → CATS

RIGHT → BIGHT → BIGOT → BIGOS → BIGGS → BIOGS → BROGS → BRIGS → BRINS → BRING → WRING → WRONG

Nie każde słowo da się przeprowadzić na inne (na przykład mogą być różnej długości).

Definicja. Graf (skierowany) to para (V, E), gdzie:

- V to zbiór **wierzchołków** (lub: **węzłów**).
- E jest relacją na V. Parę $(u, v) \in E$ nazywamy **krawędzią** z u do v.

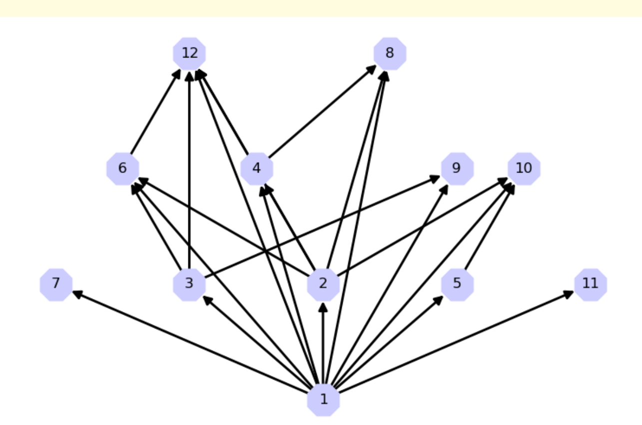
Graf nazywamy **nieskierowanym**, gdy E jest symetryczna, czyli $\forall u, v \in V\big((u, v) \in E \iff (v, u) \in E\big)$.

Inna formalizacja grafu nieskierowanego: E to zbiór nieuporządkowanych par $\{u, v\} \subset V$, gdzie $u \neq v$.

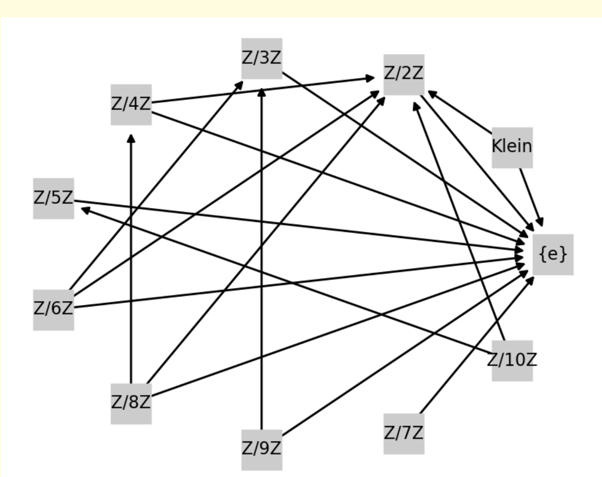
Wykład *Wstęp do matematyki*: **diagram relacji**.

Konwencja u nas: V **zawsze będzie skończony**.

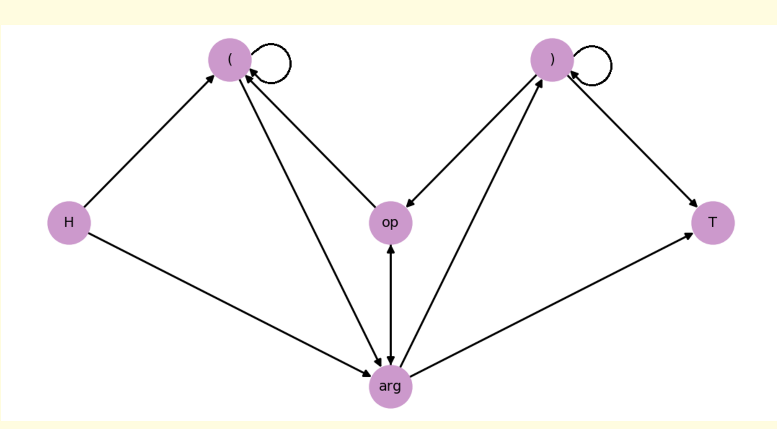
Przykład 1. $G = (V, E)$, skierowany.
$V = \{n \in \mathbb{N}_+ : n \leqslant 12\}; (n, m) \in E \iff n|m \wedge n < m$.



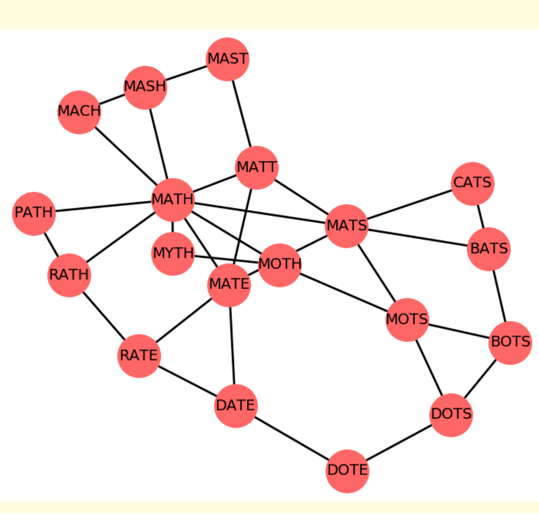
Przykład 2. $G = (V, E)$, skierowany.
$V = \{\mathbb{Z}/n\mathbb{Z} : 1 \leqslant n \leqslant 10\} \cup \{Klein\}$.
$(g_1, g_2) \in E \iff g_1 \neq g_2$ i istnieje epimorfizm $f : g_1 \to g_2$.



Przykład 3. $G = (V, E)$, skierowany.
$V = \{(,), op, arg, H, T\}$, typy tokenów.
$(t_1, t_2) \in E \iff$ token typu t_2 może wystąpić po t_1 w wyrażeniu infix.



Przykład 4. Graf (V, E) niektórych sąsiednich słów (nieskierowany).
$(w_1, w_2) \in E$ gdy słowo w_1 powstaje ze słowa w_2 przez zamianę jednej litery na inną.



Dla grafu $G = (V, E)$ przyjmujemy nazewnictwo:

- Jeśli $(u, v) \in E$, to v nazywamy **sąsiadem** u.
- **Ścieżka** (skierowana) z wierzchołka u do v to ciąg krawędzi $(u_1, v_1), \ldots, (u_n, v_n)$ takich, że $\forall i(v_i = u_{i+1})$, $u = u_1, v_n = v$.
  Uwaga: w drzewach definiowaliśmy ścieżkę inaczej!
- **Ścieżka nieskierowana** z wierzchołka u do v to ciąg krawędzi $(u_1, v_1), \ldots, (u_n, v_n)$ takich, że $\forall i\big(\{u_i, v_i\} \cap \{u_{i+1}, v_{i+1}\} \neq \varnothing\big)$, $u \in \{u_1, v_1\}, v \in \{u_n, v_n\}$.
- **Odległość** między u i v to długość najkrótszej ścieżki [lub ścieżki nieskierowanej] z u do v, jeśli taka istnieje. W przeciwnym wypadku jako odległość kładziemy ∞.

Każdy wierzchołek v jest połączony sam ze sobą pustą ścieżką. Odległość z v do v wynosi 0.

Pojęcie odległości (według ścieżek nieskierowanych) jest dobrze umotywowane teorią przestrzeni metrycznych. Niech $d\colon V \times V \to \mathbb{R} \cup \{\infty\}$ będzie funkcją zdefiniowaną przez

$$d(x, y) = \text{ odległość nieskierowana z } x \text{ do } y.$$

Fakt. Dla wszystkich $x, y, z \in V$:

1. $d(x, y) = 0 \iff x = y$,
2. $d(x, y) = d(y, x)$,
3. $d(x, z) \leqslant d(x, y) + d(y, z)$.

Fakt mówi, że d jest metryką (to, że odległości mogą być nieskończone, nie jest problemem – metryki można bez przeszkód definiować jako funkcje w $\mathbb{R} \cup \{\infty\}$). Dzięki temu możemy dla dowolnych grafów (nawet nieskończonych) stosować terminologię z przestrzeni metrycznych (kule otwarte i domknięte, średnice etc.).

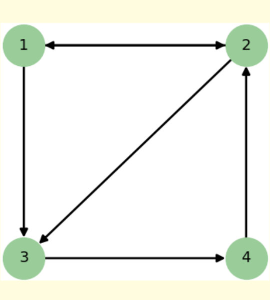
Reprezentacje grafów

Od teraz przyjmujemy, że wierzchołki grafu numerowane są (kolejnymi) dodatnimi liczbami naturalnymi, $V = \{1, 2, \ldots, n\}$.
Elementarne sposoby reprezentacji grafu:

I. Listy sąsiedztwa. Dla $i = 1, 2, \ldots n$ definiujemy listę lst_i wierzchołków, które są sąsiadami i. Złożoność pamięciowa: $\Theta(|E| + |V|)$.

II. Macierz sąsiedztwa. Definiujemy macierz $\{a_{i,j}\}_{1 \leqslant i,j \leqslant n}$, gdzie $a_{i,j} = 1$, gdy $(i, j) \in E$ i 0 w przeciwnym przypadku. Złożoność pamięciowa: $\Theta(|V|^2)$.

Przykład 5.



I. Listy sąsiedztwa:
```
1: [2, 3]
2: [1, 3]
3: [4]
4: [2]
```

II. Macierz sąsiedztwa:

$$\{a_{i,j}\} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Grafy ważone

Graf ważony to graf, w którym krawędzie mają przypisane **wagi** – liczby rzeczywiste.

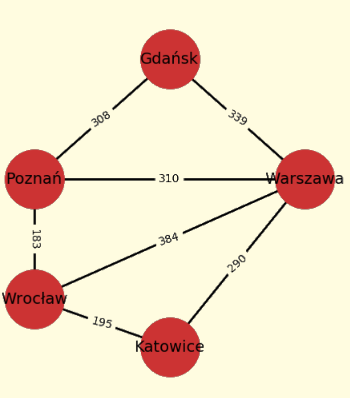
Formalnie, $G = (V, E, f)$ gdzie (V, E) jest grafem, a $f : E \to \mathbb{R}$ funkcją nadającą wagi.

(alternatywnie, $G = (V, E)$ gdzie E to funkcja częściowa z $V \times V$ w $\mathbb{R}$)

Wagi reprezentują koszt, odległość etc.

Uwaga: wagi mogą być ujemne!

Przykład 6. Graf (nieskierowany) odległości w kilometrach między niektórymi miastami w Polsce.



Uwaga. Graf $G = (V, E)$ może być nieskierowany, ale funkcja $f\colon E \to \mathbb{R}$ nie musi spełniać $f((x, y)) = f((y, x))$.

Reprezentacje grafów ważonych.
Gdańsk, Poznań, Warszawa, Wrocław, Katowice odpowiednio 1, 2, 3, 4, 5.

I. Słowniki sąsiedztwa:
```
1: {2: 308, 3: 339}
2: {1: 308, 3: 310, 4: 183}
3: {1: 339, 2: 310, 4: 384, 5: 290}
4: {2: 183, 3: 384, 5: 195}
5: {3: 290, 4: 195}
```

II. Macierz sąsiedztwa:

$$\{a_{i,j}\} = \begin{pmatrix} & 308 & 339 & & \\ 308 & & 310 & 183 & \\ 339 & 310 & & 384 & 290 \\ & 183 & 384 & & 195 \\ & & 290 & 195 & \end{pmatrix}$$

Uwaga. Brak współczynnika nie jest tożsamy z wagą 0, a z brakiem krawędzi.

Implementacja... [Pycharm]

Trawersowanie grafu – algorytm BFS

Cel: odwiedzić wierzchołki grafu w kolejności zgodnej z odległością od pewnego wierzchołka „źródłowego" t_0.

Rozróżniamy wierzchołki: „odwiedzone" (czarne), „odkryte" (szare), pozostałe (białe).

Kroki BFS (rozpoczynamy trawersowanie z wierzchołka t_0).

1. Stwórz pustą kolejkę q.

2. Oznacz t_0 jako odkryty i umieść go na końcu q.

3. Dopóki q jest niepusta:

3a. Weź wierzchołek t z początku kolejki.

3b. Oznacz nieodwiedzone, nieodkryte sąsiady t jako odkryte i umieść je na końcu q.

3c. Oznacz t jako odwiedzony.

Wierzchołki odwiedzone przez BFS poczynając od t_0 to wszystkie wierzchołki, do których prowadzi pewna (skierowana) ścieżka z wierzchołka t_0 (tzw. **silna komponenta spójności** t_0).

Fakt: Dla każdego $n \in \mathbb{N}$ istnieje krok algorytmu, w którym:

1. wszystkie wierzchołki w odległości $< n$ od t_0 są odwiedzone,
2. wszystkie wierzchołki w odległości n od t_0 są odkryte,
3. żaden wierzchołek w odległości $> n$ nie jest odkryty ani odwiedzony.

Dowód: indukcja względem n.

Wniosek: Niech u, v będą wierzchołkami, których odległości od t_0 wynoszą n, m odpowiednio. Jeśli $n < m$, to u zostanie odwiedzony przed v.
Wniosek: Niech wierzchołek u będzie w odległości n od t_0. Odwiedzając u jego nieodwiedzone, nieodkryte sąsiady są w odległości $n + 1$ od t_0.

Wniosek (jeszcze raz): Niech wierzchołek u będzie w odległości n od t_0. Odwiedzając u jego nieodwiedzone, nieodkryte sąsiady są w odległości $n + 1$ od t_0.
Uwikłamy tę obserwację w algorytm BFS.
Nowe kroki BFS (rozpoczynamy trawersowanie z wierzchołka t_0).

1. Stwórz pustą kolejkę q.

2. Oznacz t_0 jako odkryty i umieść go na końcu q.

3. Nadaj t_0 odległość 0.

4. Dopóki q jest niepusta:

4a. Weź wierzchołek t z początku kolejki.

4b. Niech $n =$ odległość nadana t.

4c. Oznacz nieodwiedzone, nieodkryte sąsiady t jako odkryte i umieść je na końcu q. Nadaj im odległość $n + 1$. Zapamiętaj, że są sąsiadami t.

4d. Oznacz t jako odwiedzony.

Implementacja... [Pycharm]

Rozwiązanie Word Ladder:

1. Zbuduj graf odpowiadający legalnym słowom, kładąc krawędzie między słowami różniącymi się jedną literą.

2. Znajdź najkrótszą scieżkę od słowa początkowego do słowa końcowego.

Implementacja [Pycharm] – dwa sposoby konstrukcji grafu (naiwny i lepszy).