

Lista 7

Przypomnienie: Grafy skierowane bez wag można reprezentować przy pomocy klasy `Graph`, przypisując każdej krawędzi wagę 1. Wszystkie rozważane grafy są skończone.

Uwaga: Dla grafu skierowanego (bez wag) G , niech $M(G)$ oznacza jego macierz sąsiedztwa.

Zadanie 1 (0,75 punktu) Niech $G = (V, E)$ będzie grafem skierowanym (bez wag).

- Opisz graf, którego macierzą sąsiedztwa jest $M(G)^T$.
- Czemu odpowiadają wyrazy macierzy $M(G)^2$?

Zadanie 2 (0,75 punktu) Niepustą ścieżkę z wierzchołka v do v nazywamy *cyklem*. Niech $G = (V, E)$ będzie grafem skierowanym, w którym dla wszystkich $u, v \in V$ zachodzi dokładnie jedno: $(u, v) \in E$, albo $(v, u) \in E$, albo $u = v$. Pokaż, że jeśli w G nie ma cyklu, to istnieje w nim wierzchołek u taki, że dla żadnego wierzchołka u' nie zachodzi $(u', u) \in E$.

Zadanie 3 (1,25 punktu) Dla grafów skierowanych bez wag $G = (V, E)$, $G' = (V', E')$ definiujemy:

- G^T – graf skierowany bez wag taki, że $M(G^T) = M(G)^T$ (*transpozycja grafu*),
- $G \cup G' := (V'', E'')$, gdzie $V'' = V \cup V'$, $E'' = E \cup E'$ (*suma grafów*).

Napisz funkcje:

- `transpose(g)`, gdzie g to graf bez wag, która zwraca transpozycję g .
- `union(g1, g2)`, gdzie $g1, g2$ to grafy bez wag, która zwraca sumę $g1$ i $g2$, utożsamiając wierzchołki grafów mające te same klucze.

Obie funkcje powinny tworzyć nowe grafy, zawierające nowe wierzchołki (obiekty typu `Vertex`) odpowiadające wierzchołkom z grafów podanych w argumentach.

Zadanie 4 (1,25 punktu) Zdefiniujmy następujące operacje na słowach:

- Zastąpienie litery w słowie inną literą.
- Usunięcie jednej litery ze słowa.
- Wstawienie jednej litery do słowa.

W pliku `collins2015.txt` (w materiałach do wykładu) znajduje się lista *legalnych słów*. Wykorzystaj wiedzę z wykładu do napisania programu, który dla podanych legalnych słów w_1 i w_2 poda najmniejszy (co do długości) ciąg operacji, które przekształcą słowo w_1 w słowo w_2 tak, aby słowo uzyskane po każdej operacji było legalne. Program powinien wypisać te operacje. Jeśli taki ciąg nie istnieje, zgłoś błąd. Przykładowe wyjście dla słów `RIGHT` i `WRONG`:

```
RIGHT → HIGHT: replace 'R' at index 0 with 'H'
HIGHT → HIGH: remove 'T' at index 4
HIGH → HOGH: replace 'I' at index 1 with 'O'
HOGH → HOG: remove 'H' at index 3
HOG → HONG: add 'N' at index 2
HONG → RONG: replace 'H' at index 0 with 'R'
RONG → WRONG: add 'W' at index 0
```