Krystian Jasionek MDL

1. Niech $A(x)$ będzie funkcją tworzącą ciągu a_n. Podaj postać funkcji tworzącej dla ciągu

   $s_n = a_0 + a_1 + a_2 + \ldots + a_n$.

   Wskazówka: Trzeba użyć funkcji tworzącej $\frac{1}{1-x}$.

Weźmy ciąg $s_n = a_0, a_1, a_2, \dots, a_n$

Wtedy jego funkcja tworząca $S(x) = \sum_{n=0}^{\infty} x^n \cdot \sum_{i=0}^{n} a_i = a_0x^0 + (a_0+a_1)x^1 + (a_0+a_1+a_2)x^2 + \dots$

Możemy wymnożyć nawiasy i pogrupować czynniki względem a, a nie x.

$$S(x) = a_0x^0 + (a_0+a_1)x^1 + (a_0+a_1+a_2)x^2 + \dots$$

$$S(x) = a_0x^0 + a_0x^1 + a_1x^1 + a_0x^2 + a_1x^2 + a_2x^2 + \dots$$

Zauważmy, że a_0 występuje przy $x^k, k \in [0, \infty]$, a_1 przy $x^k, k \in [1, \infty)$, ogólniej a_i występuje przy $x^k, k \in [i, \infty]$, więc możemy pogrupować wyrazy w następujący sposób.

$$S(x) = a_0(x^0 + x^1 + x^2 + \dots) + a_1(x^1 + x^2 + \dots) + a_2(x^2 + x^3 + \dots) + \dots$$

Chcemy, by przy każdym a stała ta sama suma x-ów. Zatem z każdego nawiasu możemy wyłączyć x w odpowiedniej potędze.

$$S(x) = a_0(x^0 + x^1 + \dots) + a_1x(x^0 + x^1 + \dots) + a_2x^2(x^0 + x^1 + \dots) + \dots$$

$$S(x) = (x^0 + x^1 + x^2 + \dots) \cdot (a_0 + a_1x + a_2x^2 + \dots)$$

Otrzymamy zatem, że

$$S(x) = \frac{1}{1-x} \cdot A(x)$$

2. Wyznacz funkcje tworzące ciągów:

   (a) $a_n = n^2$

   (b) $a_n = n^3$

   (c) $\binom{n+k}{k}$

   Wskazówka: Wszędzie przyda się funkcja tworząca $\frac{1}{1-x}$. W ostatnim podpunkcie będzie to odpowiednia potęga tej funkcji.

Spójrzmy na ciąg $b_n = n$. Znajdźmy jego funkcję tworzącą $B(x)$.

$B(x) = \sum_{n=0}^{\infty} nx^n$. Zauważmy, że nx^n przypomina pochodną wyrażenia x^n, dokładniej $(x^n)' = nx^{n-1}$. Zatem $nx^n = (x^n)' \cdot x$. Znamy funkcję tworzącą x^n, zatem możemy wyznaczyć $B(x)$.

$$B(x) = \sum_{n=0}^{\infty} nx^n = \sum_{n=0}^{\infty} (x^n)' \cdot x = \left(\frac{1}{1-x}\right)' \cdot x = \frac{x}{(1-x)^2}$$

Funkcja $B(x)$ i analogiczne rozumowanie pomoże nam obliczyć punkty (a) oraz (b).

a) $a_n = n^2$. Znajdźmy jego f. tworzącą $A^*(x)$.

$$A^*(x) = \sum_{n=0}^{\infty} n^2 x^n = \sum_{n=0}^{\infty} (nx^n)' \cdot x = \sum_{n=0}^{\infty} n^2 x^{n-1} \cdot x = \sum_{n=0}^{\infty} n^2 x^n = \sum_{n=0}^{\infty} B'(x) \cdot x$$

$$A^*(x) = \left(\frac{x}{(1-x)^2}\right)' \cdot x$$

$$A^*(x) = \frac{(1-x)^{\cancel{2}} + 2(\cancel{1-x})x}{(1-x)^{\cancel{4}\,3}} \cdot x = \frac{1-x+2x}{(1-x)^3} \cdot x$$

$$A^*(x) = \frac{(1+x)x}{(1-x)^3}$$

b) $a_n = n^3$. Znajdźmy jego funkcję tworzącą $A^{**}(x)$. Skorzystamy znów z tego samego rozumowania.

$$A^{**}(x) = \sum_{n=0}^{\infty} n^3 x^n = \sum_{n=0}^{\infty} (n^2 x^n)' \cdot x = \sum_{n=0}^{\infty} n^3 x^{n-1} \cdot x = \sum_{n=0}^{\infty} n^3 x^n = \sum_{n=0}^{\infty} (A^*(x))' \cdot x$$

$$A^{**}(x) = \left(\frac{(1+x)x}{(1-x)^3}\right)' \cdot x = \frac{(2x+1)(1-x)^{\cancel{3}} + 3(\cancel{1-x})^{\cancel{2}}(x+x^2)}{(1-x)^{\cancel{6}\,4}} \cdot x$$

$$A^{**}(x) = \frac{2x - 2x^2 + 1 - x + 3x + 3x^2}{(1-x)^4} \cdot x = \frac{(x^2+4x+1)x}{(1-x)^4}$$

c) Zastosujemy nieco odmienne podejście.

$a_n = \binom{n+k}{k}$. Znajdźmy jego funkcję tworzącą $A(x)$.

$$A(x) = \sum_{n=0}^{\infty} \binom{n+k}{k} x^n = \sum_{n=0}^{\infty} \frac{(n+k)!}{k!\,n!} x^n$$

Ale $\frac{1}{k!}$ to stała, możemy ją wyciągnąć przed sumę.

$$A(x) = \frac{1}{k!} \sum_{n=0}^{\infty} \frac{(n+k)!}{n!} x^n = \frac{1}{k!} \sum_{n=0}^{\infty} (n+k)(n+k-1)(n+k-2)\dots(n+1) x^n$$

Zauważmy, że ponownie przypomina to pochodną, dokładnie k-tą pochodną x^{n+k}, ale $\left(x^{n+k}\right)^{(k)} = (n+k)(n+k-1)\dots(n+1)\,x^{n-k}$.

Stąd

$$A(x) = \frac{1}{k!} \sum_{n=0}^{\infty} \left(x^{n+k}\right)^{(k)}$$

, ale x^k jest niezależny od n, więc możemy wyciągnąć go przed sumę.

$$A(x) = \frac{1}{k!} \left(x^k \sum_{n=0}^{\infty} x^n\right)^{(k)}$$

Wiemy, że $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$, zatem

$$A(x) = \frac{1}{k!} \left(\frac{x^k}{1-x}\right)^{(k)}$$

Policzmy $L = \left(\frac{x^k}{1-x}\right)^{(k)}$.

suma k wyrazów ciągu geometrycznego

$$L = \left(\frac{x^k}{1-x}\right)^{(k)} = \left(\frac{x^k - 1 + 1}{1-x}\right)^{(k)} = \left(\frac{x^k - 1}{1-x} + \frac{1}{1-x}\right)^{(k)} = \left[-\left(\frac{1-x^k}{1-x}\right) + \frac{1}{1-x}\right]^{(k)}$$

$$L = \left[-\left(x^0 + x^1 + x^2 + \dots + x^{k-1}\right) + \frac{1}{1-x}\right]^{(k)}$$

Zauważmy, że $\left(x^m\right)^{(n)} = 0$, gdy $n > m$, zatem

$$L = \left(0 + \frac{1}{1-x}\right)^{(k)} = \left(\frac{1}{1-x}\right)^{(k)} = \left((1-x)^{-1}\right)^{(k)}$$

Weźmy $y = (1-x)^{-1}$, wtedy

$$y' = (1-x)^{-2}$$

$$y'' = 2(1-x)^{-3}$$

$$y''' = 6(1-x)^{-4}$$

$$\vdots$$

$$y^{(k)} = k!\,(1-x)^{-(k+1)}$$

Zatem

$$A(x) = \frac{1}{k!}\cdot k!\,(1-x)^{-(k+1)}$$

$$A(x) = \frac{1}{(1-x)^{k+1}}$$

$$A(x) = \frac{1}{(1-x)^{k+1}}$$

3. Oblicz funkcje tworzące ciągów:

   (a) $a_n = n$ dla parzystych n i $a_n = 1/n$ dla nieparzystych n

   (b) $H_n = 1 + 1/2 + \ldots + 1/n$ $(H_0 = 0)$.

a) $a_n = 0 + 1 + 2 + \frac{1}{3} + 4 + \frac{1}{5} + 6 \ldots$

Niech $A(x)$ to funkcja tworząca ciągu a_n.

$$A(x) = 0x^0 + 1x^1 + 2x^2 + \frac{x^3}{3} + 4x^4 + \frac{x^5}{5} + \ldots$$

Możemy wyrazić a_n jako sumę dwóch innych ciągów $a_n^* = 0, 1, 2, 3, 4 \ldots$, $a_n^{**} = 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \ldots$, biorąc co drugi wyraz tych ciągów. Niech $A^*(x)$ i $A^{**}(x)$ to f. tworzące a_n^* i a_n^{**}. Wyznaczmy te funkcje.

$A^*(x) = 0x^0 + 1x^1 + 2x^2 + 3x^3 + \ldots = \sum_{n=0}^{\infty} nx^n$. Z zadania 2. wiemy, że

$$A^*(x) = \sum_{n=0}^{\infty} nx^n = \frac{x}{(1-x)^2}$$

Chcemy wybrać co drugi wyraz a_n^*, tzn. chcemy wszystkie liczby parzyste $0, 2, 4, 6 \ldots$. Zauważmy, że

$A^*(x^2) = \sum_{n=0}^{\infty} nx^{2n} = \frac{x^2}{(1-x^2)^2} = 0x^0 + x^2 + 2x^4 + 3x^6 + \ldots$ Zatem wystarczy użyć

$$2A^*(x^2) = 0x^0 + 2x^2 + 4x^4 + 6x^6 + \ldots = \frac{2x^2}{(1-x^2)^2}$$

$A^{**}(x) = 1x^1 + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \ldots = \sum_{n=0}^{\infty} \frac{1}{n+1}x^{n+1}$. Zauważmy, że $\int x^i\,dx = \frac{1}{i+1}x^{i+1} + C$. Stąd

$$\int \sum_{i=0}^{\infty} x^i\,dx = \int \frac{1}{1-x}\,dx = -\log(1-x) + C = \log 1 - \log(1-x) + C = \log\frac{1}{1-x} + C$$

$$A^{**}(x) = \sum_{n=0}^{\infty} \frac{1}{n+1}x^{n+1} = \sum_{n=0}^{\infty}\int_0^x x^n\,dx = \int_0^x \sum_{n=0}^{\infty} x^n\,dx$$

$$A^{**}(x) = \int_0^x \frac{1}{1-x} = \log\frac{1}{1-x} - \log\frac{1}{1} = \boxed{\log\left(\frac{1}{1-x}\right)}$$

Chcemy tylko wyrazy $1, \frac{1}{3}, \frac{1}{5}, \ldots$, czyli odwrotności nieparzystych n. Zauważmy, że

$$\frac{A^{**}(x) - A^{**}(-x)}{2} = \frac{\sum_{n=0}^{\infty}\frac{1}{n+1}x^{n+1} - \sum_{n=0}^{\infty}\frac{1}{n+1}(-x)^{n+1}}{2} = \frac{\sum_{n=0}^{\infty}\frac{1}{n+1}x^{n+1} - \frac{1}{n+1}(-x)^{n+1}}{2}$$

$$\frac{A^{**}(x) - A^{**}(-x)}{2} = \frac{1x + 1x + \frac{1}{2}x^2 - \frac{1}{2}x^2 + \frac{1}{3}x^3 + \frac{1}{3}x^3 + \frac{1}{4}x^4 - \frac{1}{4}x^4 + \ldots}{2} = \frac{2x + 2\cdot\frac{1}{3}x^3 + 2\frac{1}{5}x^5 + \ldots}{2}$$

$\frac{A^{**}(x) - A^{**}(-x)}{2} = 1x + \frac{1}{3}x^3 + \frac{1}{5}x^5 + \ldots$, czyli otrzymamy to, co chcemy.

Zatem $A(x) = 0 + 1x + 2x^2 + \frac{1}{3}x^3 + 4x^4 + \frac{1}{5}x^5 + \ldots$ możemy wyrazić jako

$$A(x) = 2A^*(x^2) + \frac{A^{**}(x) - A^{**}(-x)}{2}$$

$$A(x) = \frac{2x^2}{(1-x^2)^2} + \frac{\log\frac{1}{1-x} - \log\frac{1}{1+x}}{2} = \frac{2x^2}{(1-x^2)^2} + \frac{\log\frac{\frac{1}{1-x}}{\frac{1}{1+x}}}{2}$$

$$A(x) = \frac{2x^2}{(1-x^2)^2} + \frac{1}{2}\log\frac{1+x}{1-x}$$

b) Szukamy funkcji tworzącej dla ciągu $H_0, H_1, H_2 \ldots$, gdzie $H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$, $H_0 = 0$

Niech szukana funkcja to $H(x)$. Wtedy

$$H(x) = 0x^0 + (0+1)x^1 + (0+1+\frac{1}{2})x^2 + \ldots = \sum_{n=0}^{\infty} x^n \cdot \sum_{k=0}^{n} \frac{1}{k}$$

Zauważmy, że jest to bardzo podobna funkcja do tej z pierwszego zadania. Wykorzystajmy podobne rozumowanie, co w tamtym przykładzie.

$$H(x) = 0x^0 + 0x^1 + 1x^1 + 0x^2 + 1x^2 + \frac{1}{2}x^2 + \ldots$$

$$H(x) = 0(x^0 + x^1 + x^2 + \ldots) + 1(x^1 + x^2 + x^3 + \ldots) + \frac{1}{2}(x^2 + x^3 + \ldots) = 0(x^0 + x^1 + \ldots) + 1x(x^0 + x^1 + \ldots) + \frac{1}{2}x^2(x_0 + \ldots) + \ldots$$

$$H(x) = (0 + 1x + \frac{1}{2}x^2 + \ldots)(x^0 + x^1 + x^2 + \ldots) = \left(0 + \sum_{n=0}^{\infty} \frac{1}{n+1}x^{n+1}\right) \cdot \left(\sum_{n=0}^{\infty} x^n\right)$$

Ale $\sum_{n=0}^{\infty} \frac{1}{n+1}x^{n+1}$ policzyliśmy w punkcie (a), skąd wiemy, że wynosi on $\log\left(\frac{1}{1-x}\right)$.

$$H(x) = \left(0 + \log\frac{1}{1-x}\right)\left(\frac{1}{1-x}\right) = 0 + \frac{\log\frac{1}{1-x}}{1-x}$$

4. Niech $A(x)$ będzie funkcją tworzącą ciągu a_n. Znajdź funkcję tworzącą ciągu b_n postaci $(a_0, 0, 0, a_3, 0, 0, a_6, \ldots)$, czyli takiego, że dla każdego naturalnego k, $b_{3k} = a_{3k}$ oraz $b_{3k+1} = b_{3k+2} = 0$.

Wskazówka: Użyj zespolonych pierwiastków stopnia 3 z 1.

Wyznaczmy pierwiastki zespolone trzeciego stopnia z 1. Są to

$$x_0 = \frac{-1 + i\sqrt{3}}{2} \quad x_1 = \frac{-1 - i\sqrt{3}}{2} \quad x_2 = 1$$

Wyznaczmy $A(x_0 x)$ oraz $A(x_1 x)$

$A(x_0 x) = a_0 + a_1 x_0 x + a_2 x_0^2 x^2 + a_3 x_0^3 x^3 + a_4 x_0^4 x^4 + \ldots$, ale zauważmy, że $x_0^3 = 1$, zatem

$$A(x_0 x) = a_0 + a_1 x_0 x + a_2 x_0^2 x^2 + a_3 x^3 + a_4 x_0 x^4 + \ldots$$

$A(x_1 x) = a_0 + a_1 x_1 x + a_2 x_1^2 x^2 + a_3 x_1^3 x^3 + \ldots$, ale zauważmy, że $x_1^3 = 1$, zatem

$$A(x_1 x) = a_0 + a_1 x_1 x + a_2 x_1^2 x^2 + a_3 x^3 + \ldots$$

Sprawdźmy, co otrzymamy po zsumowaniu tych funkcji:

$$A(x_0 x) + A(x_1 x) = a_0 + a_0 + a_1 x_0 x + a_1 x_1 x + a_2 x_0^2 x^2 + a_2 x_1^2 x^2 + \ldots$$

$$A(x_0 x) + A(x_1 x) = 2a_0 + (x_0 + x_1) a_1 x + (x_0^2 + x_1^2) a_2 x^2 + 2a_3 x^3 + (x_0 + x_1) a_4 x^4 + \ldots$$

Zauważmy, że

$$x_0 + x_1 = \frac{-1 + i\sqrt{3}}{2} + \frac{-1 - i\sqrt{3}}{2} = \frac{-2}{2} = -1$$

$$x_0^2 + x_1^2 = \frac{1 - 2i\sqrt{3} - 3 + 1 + 2i\sqrt{3} - 3}{4} = \frac{-4}{4} = -1$$, zatem

$$A(x_0x) + A(x_1x) = 2a_0 - a_1x - a_2x^2 + 2a_3x^3 - a_4x^4 - a_5x^5 + \dots$$

Jeśli otrzymaną sumę dodamy do $A(x)$ otrzymamy

$$A(x) + A(x_0x) + A(x_1x) = a_0 + 2a_0 + a_1x - a_1x + a_2x^2 - a_2x^2 + a_3x^3 + 2a_3x^3 - \dots$$

$$A(x) + A(x_0x) + A(x_1x) = 3a_0 + 0 + 0 + 3a_3x^3 + 0 + 0 + 3a_6x^6 + \dots$$, zatem otrzymaliśmy funkcję $B(x)$ tworzącą ciąg $b_n = a_0, 0, 0, a_3, 0, 0, a_6, \dots$

$$B(x) = A(x) + A(x_0x) + A(x_1x)$$, gdzie $x_0 = \frac{-1 + i\sqrt{3}}{2}$, $x_1 = \frac{-1 - i\sqrt{3}}{2}$

6. Niech Q_k oznacza graf k-wymiarowej kostki, tzn. zbiór wierzchołków tego grafu tworzą wszystkie k-elementowe ciągi zer i jedynek i dwa wierzchołki są sąsiednie wtedy i tylko wtedy, gdy odpowiadające im ciągi różnią się dokładnie jedną współrzędną. Oblicz, ile wierzchołków i krawędzi ma graf Q_k.

Niech V_k - zbiór wierzchołków kostki k-wymiarowej, E_k - zbiór krawędzi tej kostki.

Reprezentujemy elementy V_k jako k-elementowe ciągi zer i jedynek. Mamy zatem k elementów, gdzie każdy może przyjąć dwie wartości, zatem takich ciągów jest 2^k, a stąd $|V_k| = 2^k$.

Dwa wierzchołki są sąsiadami tylko, gdy reprezentujące je ciągi różnią się na jednej współrzędnej. Weźmy dowolny $v \in V_k$. Jest on reprezentowany przez ciąg 0,1 długości k. Wybierzmy dowolnego $w \in V_k$ t. że ciąg reprezentujący w różni się od ciągu reprezentującego v na jednej pozycji. Taki ciąg ma k pozycji, zatem miejsce, w którym v i w się różnią możemy wybrać na k sposobów. Zatem każdy wierzchołek $v \in V_k$ ma k sąsiadów, zatem sumaryczna liczba sąsiadów (stopni wierzchołków) to $2^k \cdot k$. Zauważmy jednak, że każdych dwóch sąsiadów łączy jedna, ta sama krawędź, zatem łączna liczba krawędzi jest dwa razy mniejsza od sumy stopni wierzchołków. Jest to również konsekwencja twierdzenia z wykładu, że

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Zatem $|E| = 2^k \cdot k \cdot \frac{1}{2} = 2^{k-1} \cdot k$.

Mamy więc 2^k wierzchołków i $2^{k-1} \cdot k$ krawędzi.

7. Problem izomorfizmu dwóch grafów jest trudny. Załóżmy natomiast, że w komputerze są dane dwa grafy G i H, określone na tym samym zbiorze wierzchołków $V(G) = V(H) = \{1, 2, 3, \ldots, n\}$. Niech m oznacza liczbę krawędzi grafu G. Podaj algorytm sprawdzający w czasie $O(m+n)$, czy te grafy są identyczne.

1 2 3 4

N[2]=1

1: [2, 3]

2: [1, 4]

Wiemy, że zbiory wierzchołków grafów G i H są identyczne, zatem chcemy sprawdzić, czy odpowiadające sobie wierzchołki w tych grafach mają tych samych sąsiadów. Potrzebujemy w tym celu czterech tablic:

- **V** - tablica wierzchołków (są takie same dla grafów G i H),
- **G** - tablica list sąsiedztwa konkretnych wierzchołków w grafie G, ~~**G[v][g]** = 1 oznacza, że **v** ma w G sąsiada **g**.~~
- **H** - tablica list sąsiedztwa konkretnych wierzchołków w grafie H, ~~**H[v][h]** = 1 oznacza, że **v** ma w H sąsiada **h**.~~
- **N** - tablica pomocnicza długości tablicy **V**, wypełniona zerami, której zadanie omówię za chwilę.

Potrzebujemy również zmiennej pomocniczej **i** = 0.

Chcemy dla każdego wierzchołka **v** w **V** sprawdzić czy jego listy sąsiadów w grafie G i w grafie H są takie same, tzn. czy zawierają tę samą liczbę sąsiadów i czy są to ci sami sąsiedzi. W tym celu dla każdego **v** w **V** przeszukujemy listę **G[v]**. Tutaj przyda nam się tablica **N**, tzn. dla każdego sąsiada **g** w **G[v]** zaznaczamy go w **N** poprzez ustawienie wartości **N[g]** na 1. Zwiększamy również licznik sąsiadów **i** o 1.

Gdy przeszukamy całą listę **G[v]** przechodzimy do **H[v]** i sprawdzamy, czy dla któregoś sąsiada **h** w **H[v]** wartość **N[h]** == 0, to by oznaczało, że w H wierzchołek **v** ma sąsiada, którego nie ma w G i grafy nie są identyczne. Możemy wtedy zakończyć algorytm, zwracając fałsz. W przeciwnym wypadku ustawiamy **N[h]** na 0 (by oczyścić tablicę dla sprawdzenia sąsiadów kolejnego wierzchołka z **V**) i zmniejszamy licznik sąsiadów **i** o 1.

Po sprawdzeniu wszystkich elementów **H[v]** sprawdzamy, czy **i** == 0. Jeśli nie, to znaczy, że liczba sąsiadów wierzchołka **v** jest różna w obu grafach, czyli grafy te nie są identyczne. W takim wypadku zwracamy fałsz.

Wszystkie te czynności powtarzamy dla każdego wierzchołka **v** z **V**. Jeśli algorytm przebiegł bez problemów, zwracamy prawdę, bo G i H są identyczne.

Możemy to zapisać w postaci algorytmu:

```
V = [n]
G = [n][deg(v)]
H = [n][deg(v)]
N = [0] * n

i = 0

for v in V:
    for g in G[v]:
        N[g] = 1
        i += 1

    for h in H[v]:
        if N[h] == 0:
            return False
        N[h] = 0
        i -= 1

    if i != 0:
        return False

return True
```

Sprawdźmy, jaką złożoność czasowa ma zaprezentowany algorytm. Mamy n wierzchołków $v_1, v_2, v_3, \ldots, v_n$. Dla każdego z nich musimy sprawdzić wszystkich jego sąsiadów w obu grafach. W takim razie pętla po elementach z **G** wykonuje tyle obrotów, ile wynosi liczba sąsiadów wierzchołka. Nie wiemy, ilu sąsiadów ma wierzchołek **v** w **H**, ale zauważmy, że w pesymistycznym scenariuszu spotkamy tych samych sąsiadów, co w **G[v]** i najwyżej jeszcze jednego, bo gdy tylko spotkamy element **h**, który nie należy do **G**, to zwracamy fałsz. Liczba sąsiadów wierzchołka $v_i = deg(v_i)$. Zatem łącznie mamy $2deg(v_1) + 2deg(v_2) + \ldots + 2deg(v_n) + 1 = 2\sum_{i=1}^{n} deg(v_i) + 1$ operacji. Z wykładu wiemy, że $\sum_{i=1}^{n} deg(v_i) = 2|E|$, zatem mamy $4m+2$ operacji. Ale jeśli lista sąsiadów każdego n jest pusta, wtedy wykonuje się jedynie pętla wewnętrzna, zatem mamy n operacji.
Zatem złożoność czasowa tego algorytmu to $O(n + 4m + 2) = O(n+m)$.