

# Teoria Grafów - projekt zaliczeniowy

---

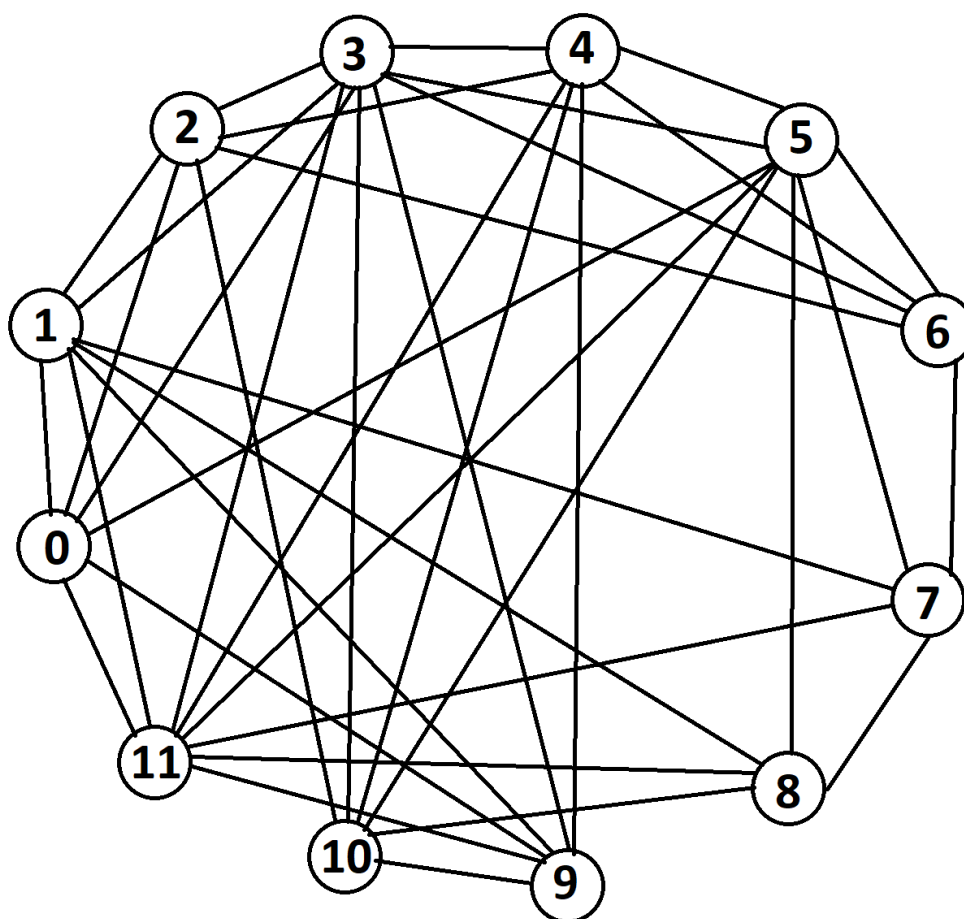
Zadania dla: Konrad Prasziewicz

## Część analityczna

W załączniku, w pliku Konrad\_Prasziewicz.json znajduje się lista sąsiedztwa dla grafu do przeanalizowania. Zadania w części analitycznej (1-8) mają zostać wykonane w oparciu o ten właśnie graf. Zadania mogą być rozwiązane na kartce i zeskanowane lub wykonane w dowolnym programie, np. OneNote. Proszę o wyniki w formie pliku pdf.

### Zadanie 1 (1pkt)

Wykonaj szkic grafu.



### Zadanie 2 (1pkt)

Opisz graf w formie macierzy incydencji.

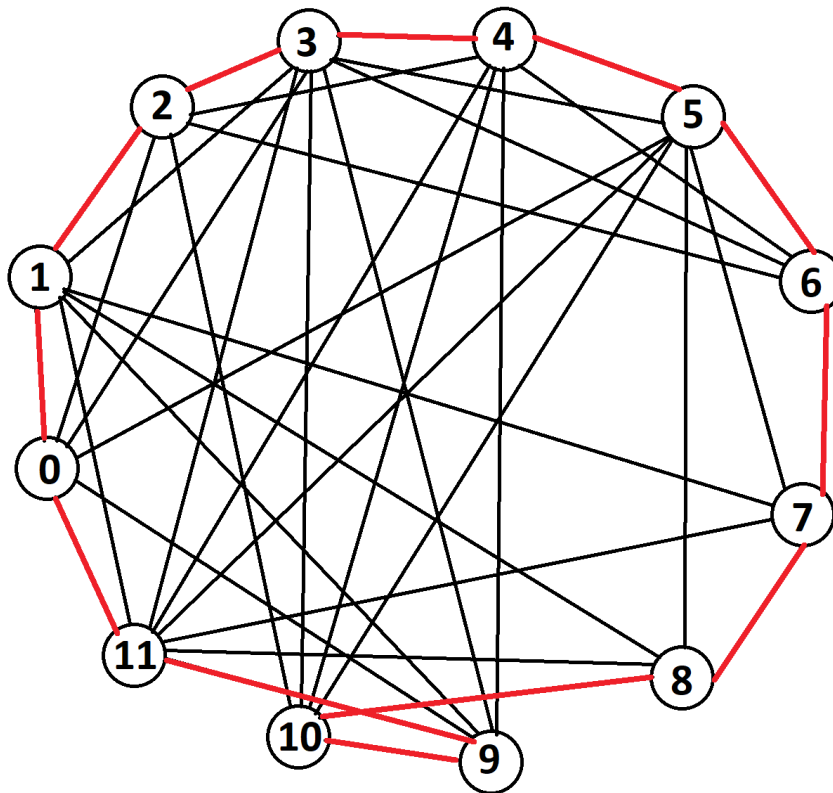
***Lista krawędzi znajduje się w pliku krawędzie.txt.***

[illegible]

### Zadanie 3 (3pkt)

Czy ten graf jest hamiltonowski/pół-hamiltonowski? Jeśli tak to podaj ścieżkę/cykl Hamiltona.

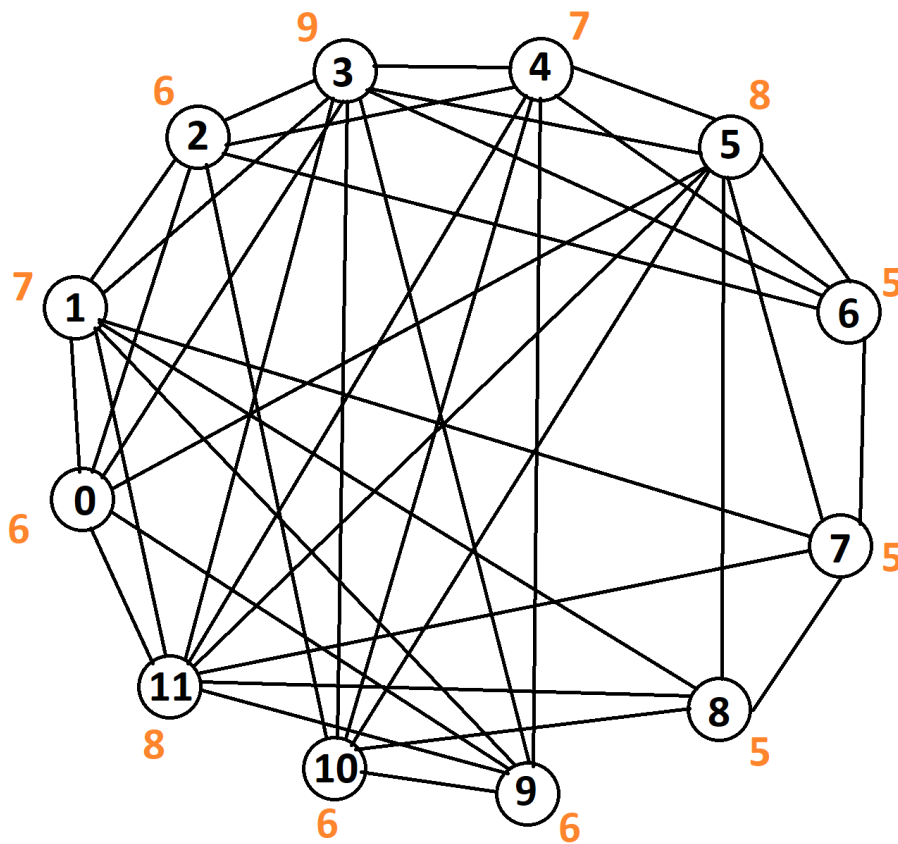
*Graf jest hamiltonowski*



#### Zadanie 4 (3pkt)

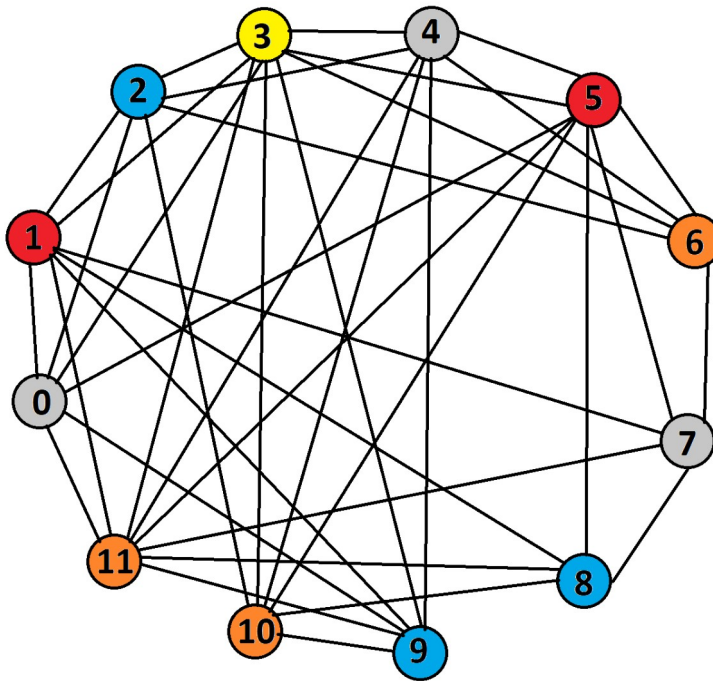
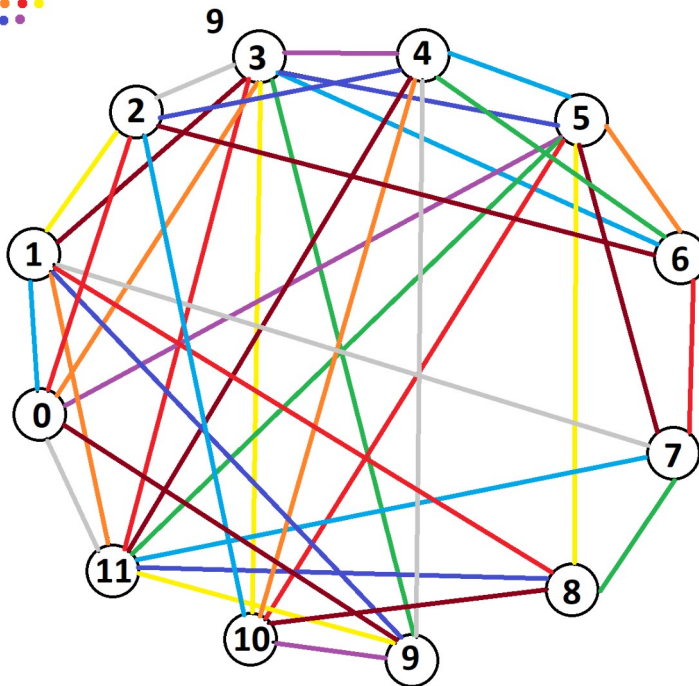
Czy ten graf jest eulerowski/pół-eulerowski? Jeśli tak to podaj ścieżkę/cykl Eulera.

***Ponad 2 wierzchołki mają nieparzysty stopień  $\Rightarrow$  graf nie jest eulerowski ani pół-eulerowski.***



### Zadanie 5 (2pkt)

Pokoloruj graf wierzchołkowo oraz krawędziowo.



### Zadanie 6 (1pkt)

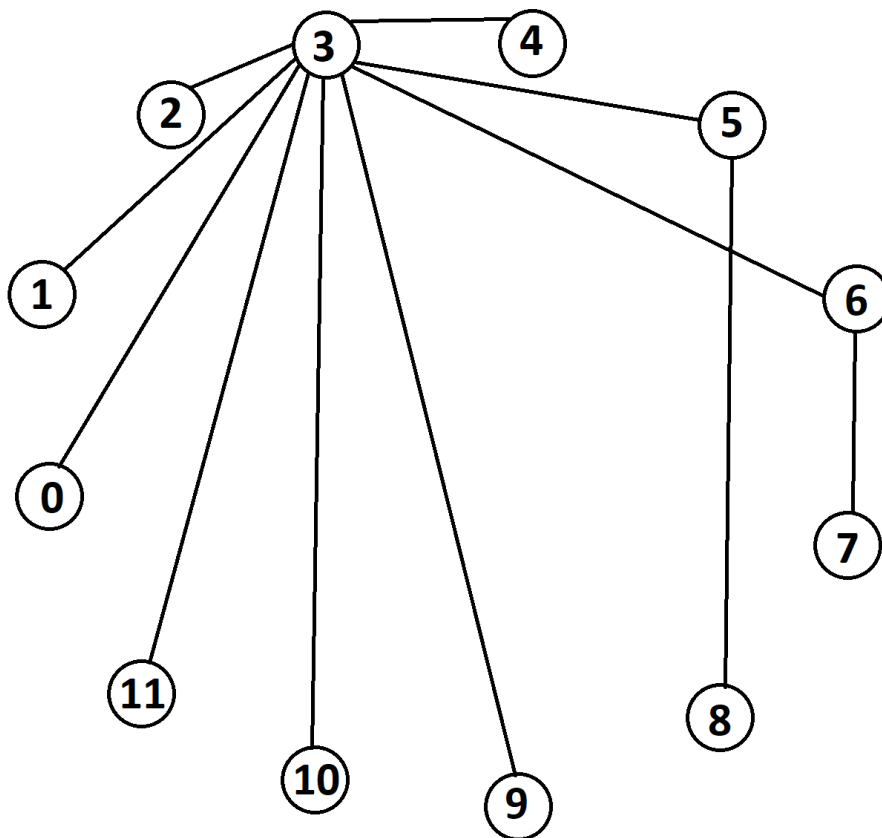
Podaj liczbę chromatyczną oraz indeks chromatyczny dla grafu.

*liczba chromatyczna: 5*

*indeks chromatyczny: 9*

### Zadanie 7 (1pkt)

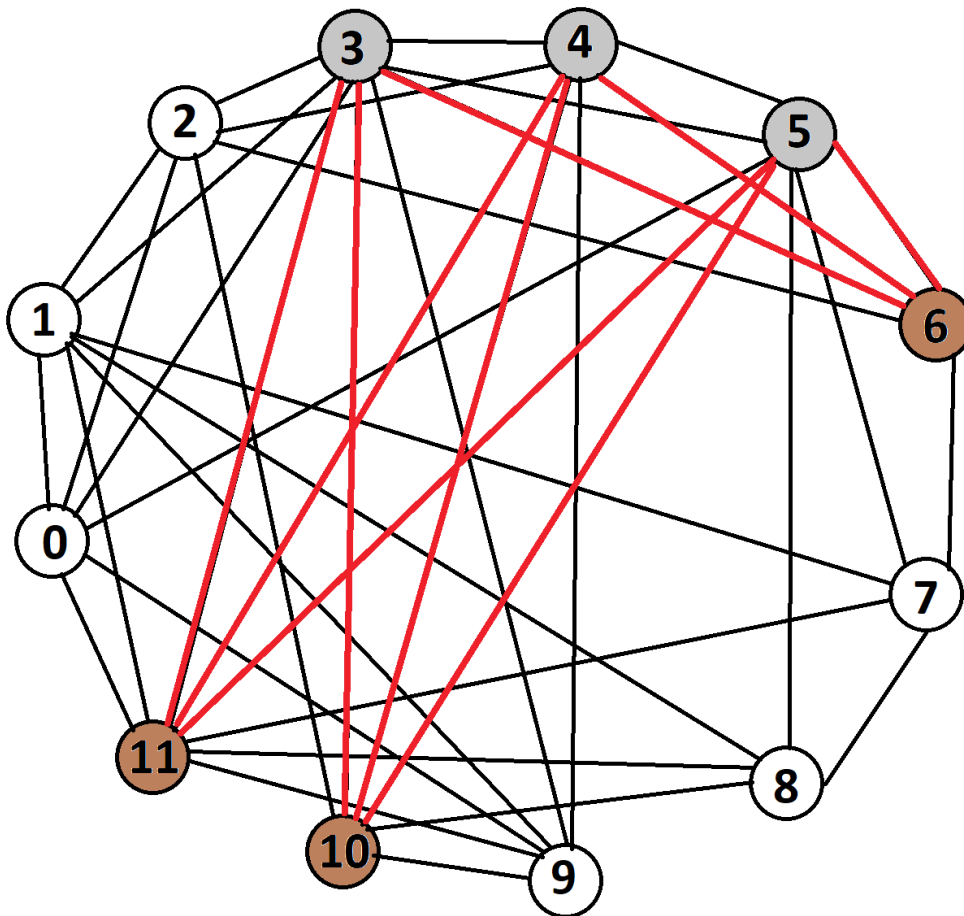
Wyznacz minimalne drzewo rozpinające dla analizowanego grafu.



### Zadanie 8 (2pkt)

Czy rysunek tego grafu jest planarny? Jeśli nie, to czy da się go przedstawić jako planarny? Jeśli tak, to ile ścian można w nim wyznaczyć? Proszę to wykazać na rysunku.

*Graf nie może być planarny, ponieważ zawiera podgraf  $K_{3,3}$ .*



## Część programistyczna

### Zaimplementuj poniższy algorytm w wybranym języku.

Algorytm może zostać zaimplementowany w wybranym języku - Java, Kotlin, C, C++, Python, JS, TS, C#. Implementację proszę dostarczyć w formie linku do repozytorium (GitHub, GitLab - preferowane) lub archiwum zip. Program ma wczytywać graf z pliku (lista sąsiedztwa bądź macierz incydencji), a następnie uruchomić zaimplementowany algorytm na tym grafie. W repozytorium musi znajdować się instrukcja uruchomienia projektu.

### Zaimplementuj algorytm Dijkstry (10pkt)

Przeanalizuj powyższy algorytm: jakie problemy rozwiązuje, konkretne przykłady wykorzystania, z jakich metod korzysta się obecnie do rozwiązywania tych problemów (4pkt)