Ćwiczenie lab. **Prawidłowe kolorowanie wierzchołków i krawędzi grafu**

Podstawowy model grafu w postaci zbioru wierzchołków i zbioru krawędzi często nie wystarcza do opisu skomplikowanych zależności i powiązań, jakie pojawiają się w praktycznych problemach. Z tego powodu wzbogaca się grafy o dodatkowe elementy, takie jak: kierunek krawędzi, waga krawędzi lub wierzchołka, kolor krawędzi lub wierzchołka itp.

W przypadku najprostszego rodzaju kolorowania grafu, każdemu wierzchołkowi albo każdej krawędzi przyporządkowuje się dokładnie jeden kolor. W zależności od problemu, ten sam kolor może być użyty tylko raz (tylko dla jednego wierzchołka lub krawędzi) albo wiele razy. W programach komputerowych, dla wygody programisty, kolory zazwyczaj oznacza się liczbami naturalnymi.

Podstawowym (najczęściej rozważanym) rodzajem kolorowania grafu jest kolorowanie prawidłowe (ang. proper coloring) jego wierzchołków albo krawędzi.

Przez kolorowanie prawidłowe wierzchołków grafu rozumie się takie przyporządkowanie każdemu wierzchołkowi dokładnie jednego koloru, że żadne dwa wierzchołki połączone krawędzią nie mają tego samego koloru. Innymi słowy, każda krawędź ma oba końce w różnych kolorach.

Przez kolorowanie prawidłowe krawędzi rozumiemy natomiast takie kolorowanie, w którym każde dwie krawędzie posiadające wspólny wierzchołek mają różne kolory. Oznacza to, że dla danego wierzchołka każda krawędź z nim incydentna musi mieć inny kolor. Tak więc, minimalna liczba kolorów potrzebna do prawidłowego pokolorowania krawędzi grafu jest nie mniejsza niż najwyższy stopień wierzchołka w tym grafie, tzn. $I(G) \geq \Delta(G)$. Vizing wykazał, że także $I(G) \leq \Delta(G) + 1$. Symbol $I(G)$ oznacza indeks chromatyczny grafu, czyli minimalną liczbę kolorów potrzebną do prawidłowego pokolorowania jego krawędzi. Liczbę tę nazywa się też krawędziową liczbą chromatyczną i czasem oznacza przez $\chi'(G)$ lub $\chi_e(G)$.

Ponieważ krawędzie wielokrotne (równoległe) mają wpływ na kolorowanie krawędzi, dlatego dla multigrafów obowiązuje inne oszacowanie indeksu chromatycznego niż dla grafów prostych, mianowicie dla multigrafów $I(G) \leq \Delta(G) + \mu(G)$, gdzie $\mu(G)$ oznacza maksymalną wielokrotność krawędzi równoległych w multigrafie. W przypadku grafów prostych $\mu(G) = 1$ i otrzymujemy oszacowanie z góry podane w poprzednim akapicie.

König udowodnił, że w grafach dwudzielnych zachodzi równość $I(G) = \Delta(G)$.

Minimalną liczbę kolorów potrzebną do prawidłowego pokolorowania wierzchołków grafu nazywa się wierzchołkową liczbą chromatyczną lub po prostu liczbą chromatyczną i oznacza $\chi(G)$.

Wyznaczanie liczby chromatycznej grafu w oparciu o różne twierdzenia ograniczające tę liczbę z dołu i z góry pozwala uzyskać dokładną wartość tej liczby jedynie dla grafów o kilku wierzchołkach. Im większa liczba wierzchołków grafu, tym ta metoda daje szerszy przedział, do którego należy liczba chromatyczna. W związku z tym potrzebne było opracowanie innej, lepszej metody, wyznaczającej dokładną, a nie przybliżoną (z przedziału) wartość $\chi(G)$.

Uniwersalnym narzędziem matematycznym, służącym do dokładnego wyznaczania liczby chromatycznej grafu, jest wielomian chromatyczny $P_G(k)$. Można z niego obliczyć liczbę prawidłowych kolorowań wierzchołków grafu za pomocą k kolorów lub mniej, czyli za pomocą co najwyżej k kolorów. Wielomian chromatyczny, dla zadanego k, zlicza więc również te kolorowania prawidłowe, w których nie użyto wszystkich k kolorów. Takie kolorowania są możliwe tylko wtedy, gdy użyto więcej kolorów niż wynosi liczba chromatyczna, a więc gdy $k > \chi(G)$.

Należy również zaznaczyć, że wielomian chromatyczny zwraca liczbę kolorowań grafu oznakowanego, tzn. takiego, w którym wierzchołki są zaetykietowane (ponumerowane) unikalnymi oznaczeniami, np. liczbami od 1 do liczby wierzchołków n. Jeżeli więc, po zaniedbaniu numerów wierzchołków, dwa kolorowania są jednakowe (pokolorowane grafy są izomorficzne), to i tak wielomian chromatyczny potraktuje je jako różne, czyli policzy jako dwa różne kolorowania. Wielomian chromatyczny bada się w dziale matematyki, zwanym algebraiczną teorią grafów.

Aby w sposób systematyczny zliczać kolorowania grafu niezaetykietowanego, należy zastosować teorię grup permutacji, która uwzględnia symetrie grafu, określane mianem automorfizmów. Nie chodzi tu o symetrie jego reprezentacji geometrycznej, chociaż niektóre automorfizmy rzeczywiście można zilustrować symetriami środkowymi albo osiowymi jego reprezentacji geometrycznej.

Najmniejsza wartość całkowita dodatnia *k*, dla której wielomian chromatyczny przyjmuje wartość niezerową (większą od zera) jest równa liczbie chromatycznej tego grafu.

Istnieją dwie uniwersalne dualne graficzne metody wyznaczania wielomianu chromatycznego. Pierwsza z nich polega na tym, że dla danego grafu G tworzy się dwa nowe grafy G_1 i G_2. Pierwszy graf posiada dodatkową krawędź, której nie było w grafie G, natomiast drugi graf powstaje z G poprzez złączenie ze sobą obu wierzchołków, między którymi dodano krawędź. Krawędzie równoległe, powstałe w wyniku procedury łączenia ze sobą wierzchołków, zastępuje się pojedynczą krawędzią, a pętle pomija się.

Druga metoda też tworzy dwa nowe grafy, ale polega na usunięciu wybranej krawędzi i złączeniu wierzchołków, które były końcami tej krawędzi. Dla grafów, które powstaną, można znowu stosować tę procedurę.

W celu wyznaczenia wielomianu chromatycznego grafu, można obie opisane metody stosować równocześnie (tzn. raz jedną, raz drugą), w zależności od tego, która prowadzi szybciej do grafów o wielomianach znanych z tabeli.

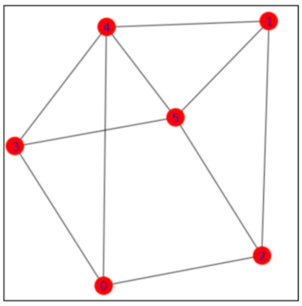
Uruchomić program Maxima i za pomocą opcji Wczytaj Pakiet (Ctrl+L) załadować bibliotekę kombinatoryczno-grafową. W pliku pdf z dokumentacją tej biblioteki znajdują się funkcje przeznaczone do badań grafów nieskierowanych w kontekście kolorowania wierzchołków oraz krawędzi. Zapoznać się z opisem tych funkcji. Załadować także standardową bibliotekę programu Maxima z funkcjami analizy grafów, za pomocą instrukcji load(graphs).

1. Wyznaczanie wielomianu chromatycznego grafu spójnego

Poniższe instrukcje dla programu Maxima losują graf spójny o zadanej liczbie wierzchołków i krawędzi. Przykładowe wygenerowane grafy pokazano niżej. W razie potrzeby można przepisać te instrukcje do Maximy i wygenerować graf o siedmiu lub większej liczbie wierzchołków. Nauczyciel przydziela każdej grupie komputerowej jeden, inny niż u pozostałych, graf o sześciu lub siedmiu wierzchołkach taki, aby wyznaczenie dla niego wielomianu chromatycznego nie było zbyt łatwe.

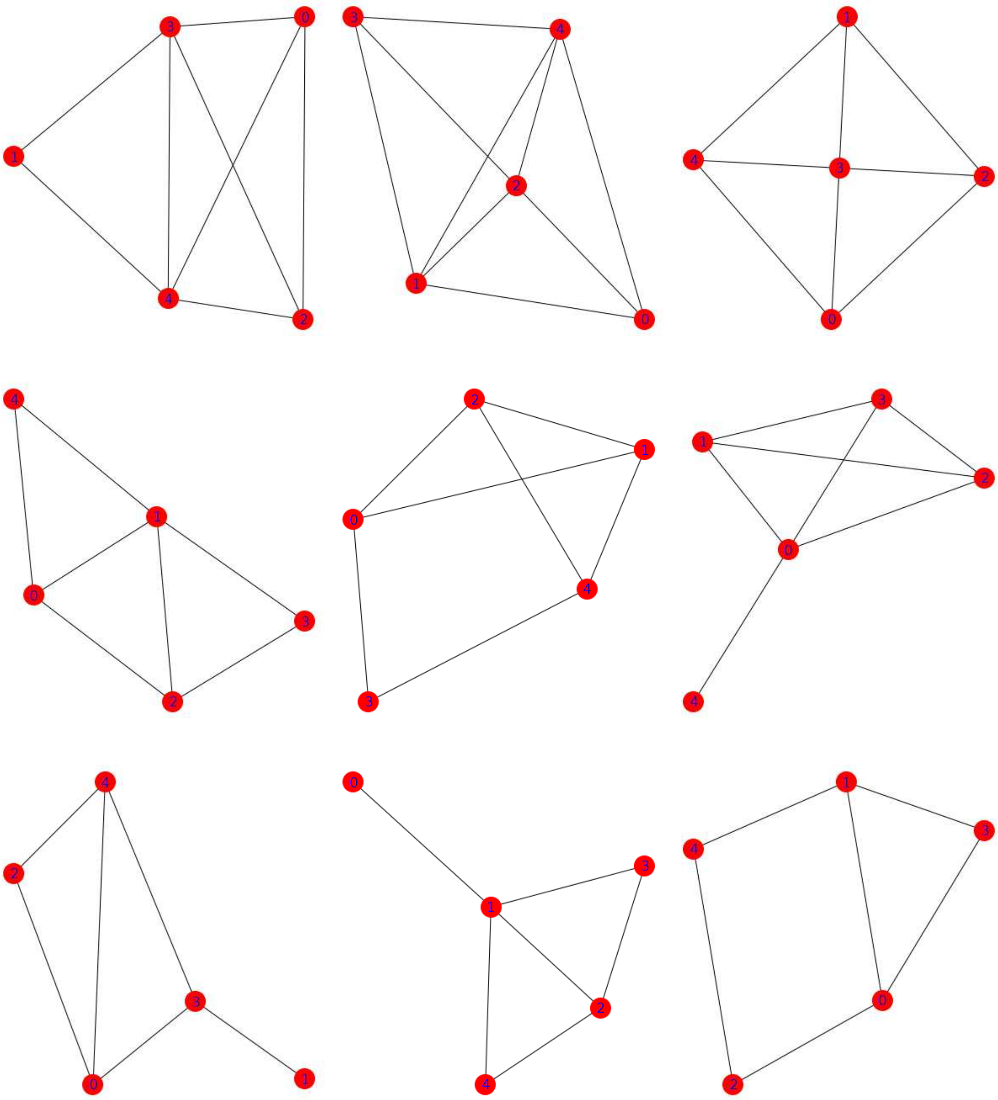
```
(%i111) do
          (
            liczba_wierzcholkow : 6,   liczba_krawedzi : 10,
            graf : random_graph1(liczba_wierzcholkow, liczba_krawedzi),
            if is_connected(graf) then return(true)
          )$
        draw_graph(graf, show_id=true, vertex_color=red, vertex_size=4);
```

(%t111)

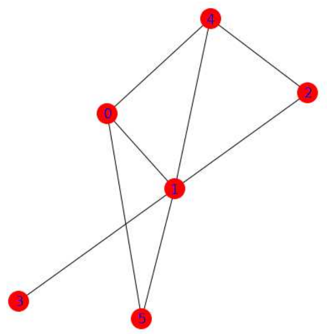
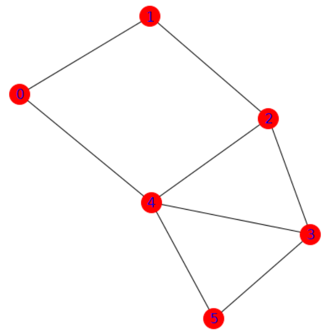
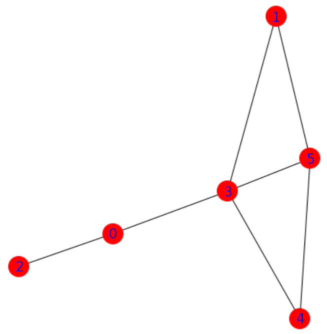
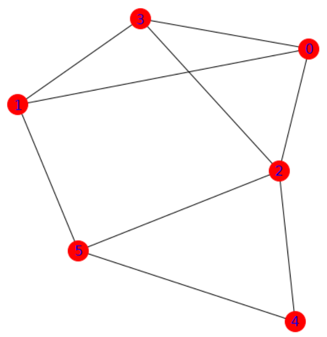
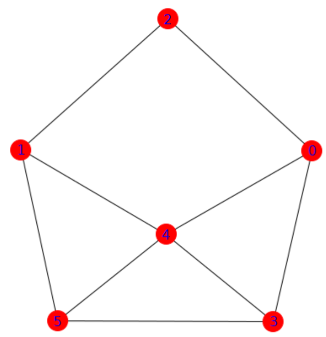
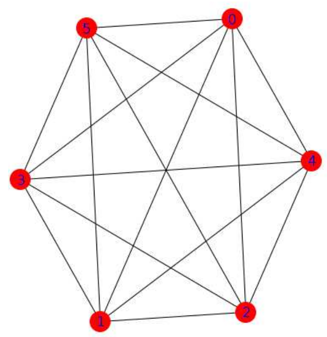
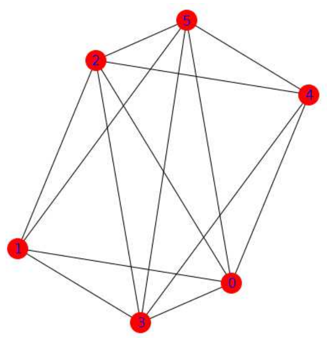
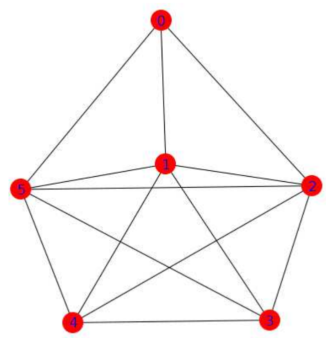
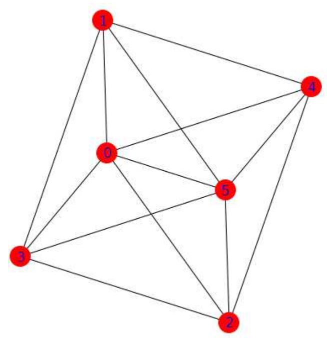
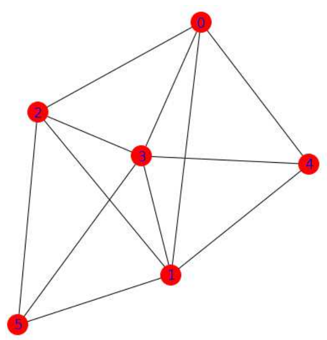
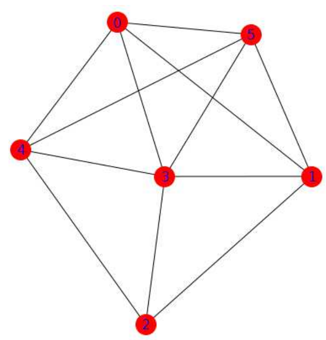
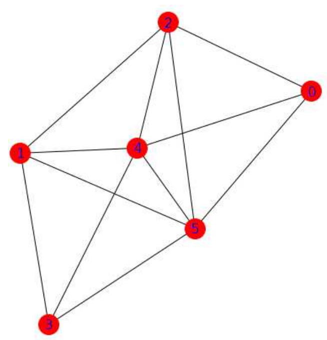
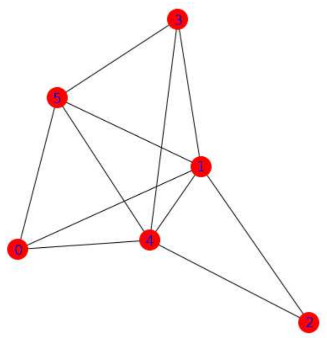
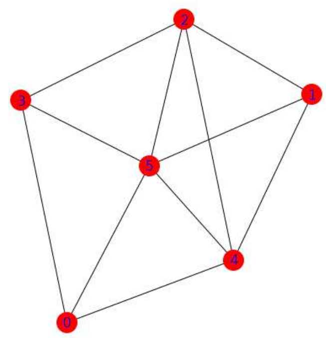
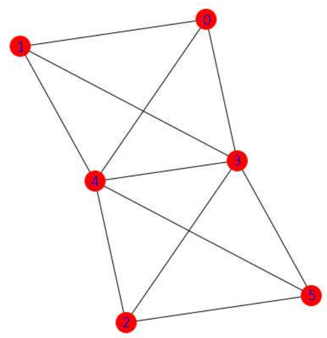


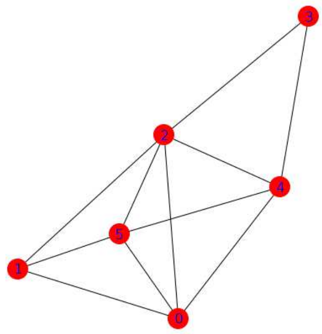
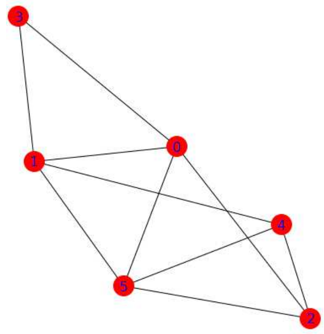
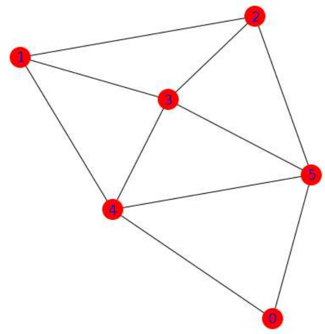
(%o111) *done*

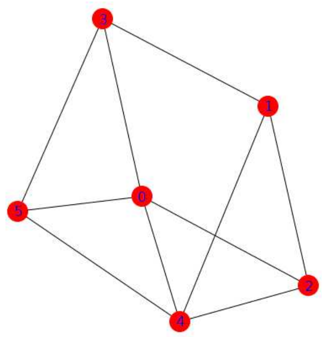
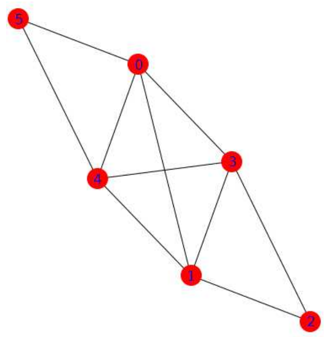
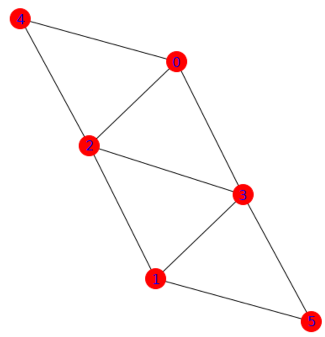
Przykładowe grafy o pięciu wierzchołkach:

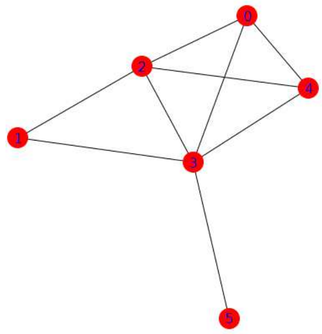
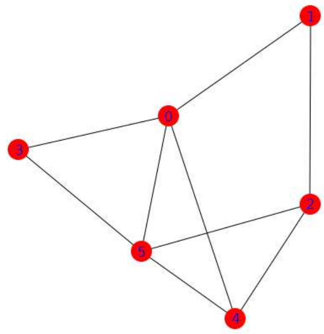
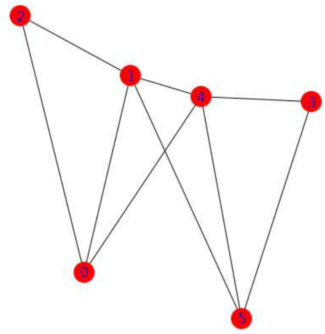


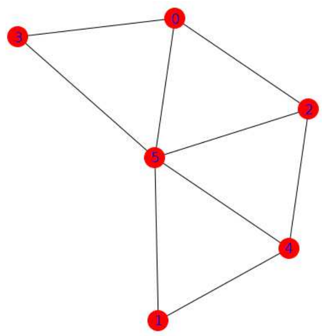
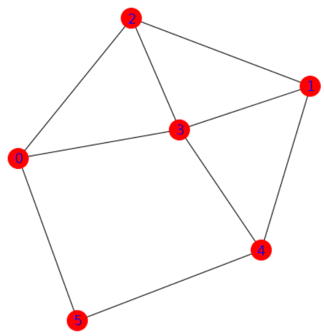
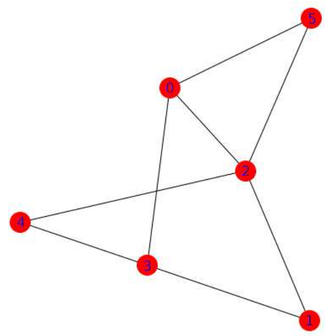
Przykładowe grafy o sześciu wierzchołkach:

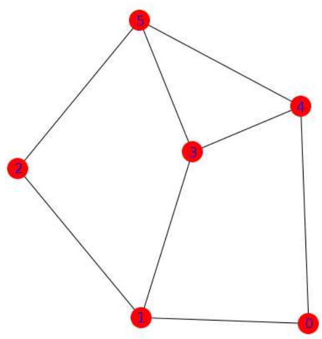
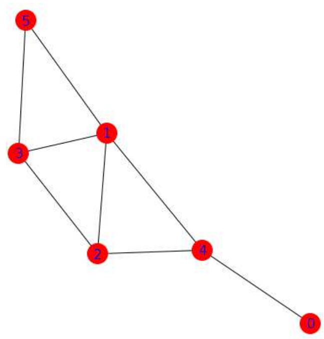
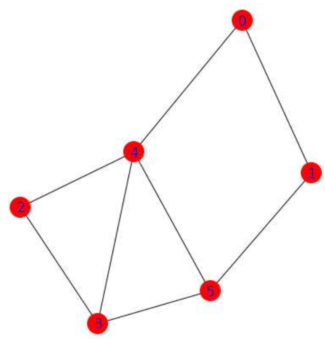
















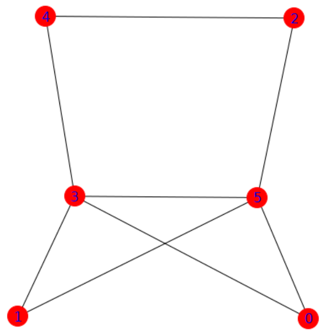
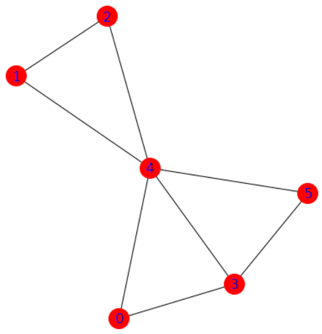
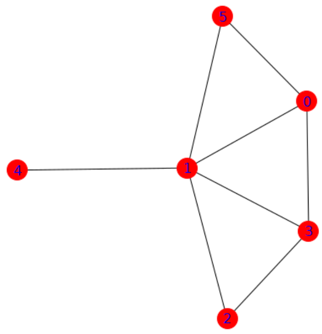
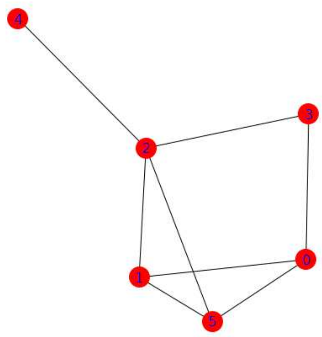
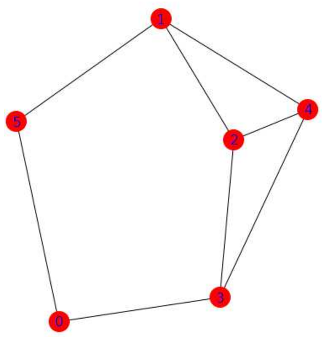
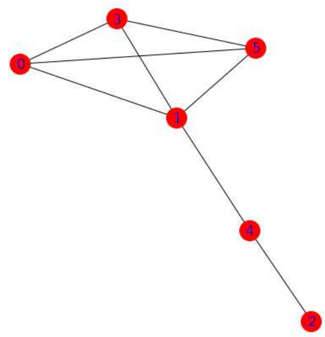
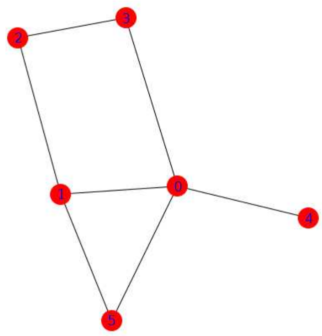
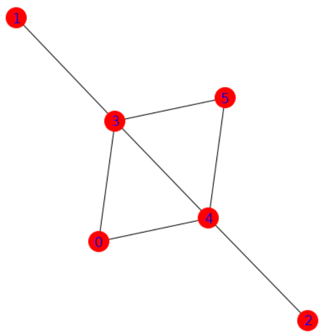
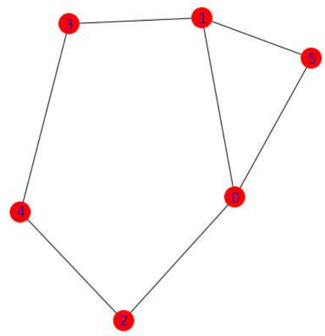
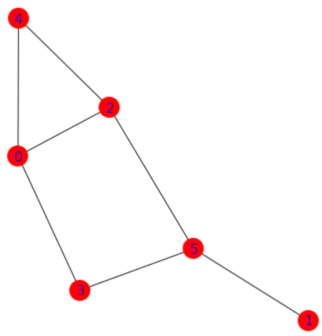
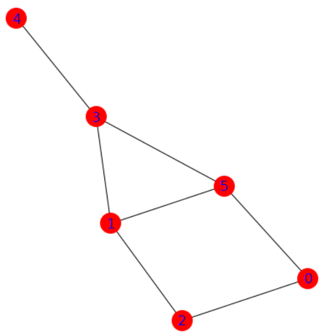
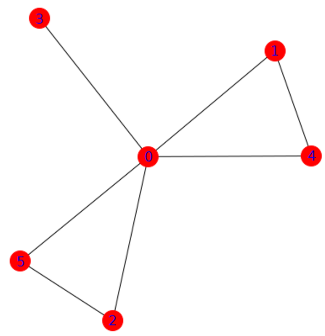




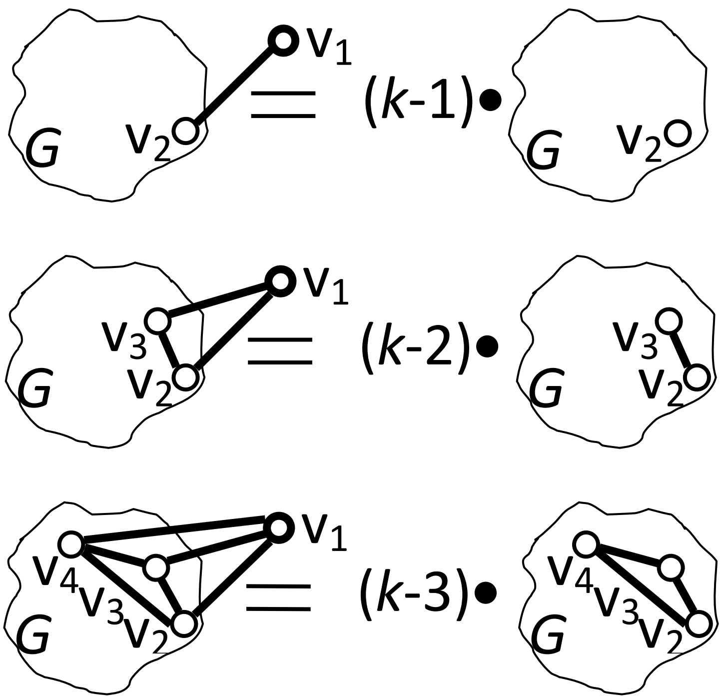




Wyznaczyć ręcznie (na kartce papieru) dwoma sposobami wielomian chromatyczny grafu przydzielonego przez nauczyciela. Najpierw metodą dodawania krawędzi, a następnie metodą usuwania krawędzi. Nie mieszać obu tych metod! Aby zminimalizować liczbę grafów pośrednich, tam, gdzie to tylko możliwe, należy stosować reguły: $k-1$, $k-2$ i $k-3$, zilustrowane poniżej.



W metodzie usuwania krawędzi trzeba pamiętać o nawiasach, gdyż między grafami będzie występował znak minus! Obie metody muszą prowadzić do tego samego wielomianu! Rysować pośrednie grafy do momentu, aż wyjdą grafy, dla których wielomiany są znane. W metodzie dodawania krawędzi należy dojść do grafów pełnych (ewentualnie do grafu kołowego), natomiast w metodzie usuwania krawędzi trzeba dojść do drzewa, cyklu, grafu pełnego lub kołowego.

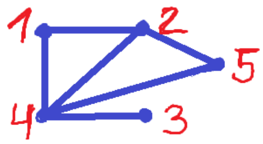
Zapisać wielomian chromatyczny, wyprowadzony dla oryginalnego grafu, w postaci rozwiniętej, tzn. w postaci, w której zazwyczaj zapisuje się wielomiany (bez nawiasów i z uproszczonymi wyrazami podobnymi). Uporządkować jednomiany według malejących potęg zmiennej *k*. Następnie przepisać poniższe instrukcje i wyznaczyć wielomian chromatyczny grafu funkcją chromatic_polynomial() z biblioteki grafowej Maximy (funkcja ta nie ma opisu w systemie pomocy!). Funkcją makelist() wyznaczyć liczbę kolorowań za pomocą liczby kolorów zmieniającej się od 0 do 10. Na podstawie tej listy wyznaczyć liczbę chromatyczną! Graf należy najpierw zdefiniować za pomocą funkcji create_graph(), której opis znajduje się w systemie pomocy programu Maxima.

```
(%i57)  wiel(k):=expand(chromatic_polynomial(graf,k))$
        wiel(k); factor(wiel(k));
        makelist(wiel(k),k,0,10);
(%o55)  k^5-6 k^4+13 k^3-12 k^2+4 k
(%o56)  (k-2)^2 (k-1)^2 k
(%o57)  [0,0,0,12,144,720,2400,6300,14112,28224,51840]
```

W tym przypadku liczba chromatyczna grafu wynosi 3, a liczb kolorowań wierzchołków grafu za pomocą co najwyżej trzech kolorów wynosi 12. Ponieważ nie da się prawidłowo pokolorować wierzchołków tego grafu za pomocą liczby kolorów mniejszej niż 3, więc liczba 12 jest równocześnie liczbą kolorowań wierzchołków tego grafu za pomocą dokładnie trzech kolorów. Podać przykłady trzech prawidłowych kolorowań wierzchołków za pomocą liczby kolorów równej liczbie chromatycznej. Kolory na rysunku grafu, obok wierzchołków, oznaczyć kolejno liczbami 1, 2, 3. Użyć tyle liczb, ile potrzeba.

Wyznaczyć wielomian chromatyczny tego grafu ponownie, ale za pomocą funkcji chromatic_poly() z biblioteki kombinatoryczno-grafowej. Funkcja ta przyjmuje graf w postaci tylko listy krawędzi i gdy wierzchołki są numerowane od 1, stąd konieczne wprowadzenie zmiennej p_graf.

```
(%i65)    p_graf:edges(graf)+1;
          chromatic_poly(p_graf,k);
(p_graf)  [[1,2],[1,4],[2,4],[2,5],[3,4],[4,5]]
(%o65)    k^5-6 k^4+13 k^3-12 k^2+4 k
```




```
chromatic_poly(graf, k)      - wyznacza zwyczajny wielomian chromatyczny gra-
                             fu, z którego można obliczyć liczbę prawidłowych
                             kolorowań zaetykietowanych wierzchołków grafu za
                             pomocą co najwyżej k kolorów. Jeżeli k jest
                             liczbą, to funkcja zwraca wartość wielomianu.
```

Wielomian chromatyczny zmiennej k grafu G, w wersji rozwiniętej $P_G(k) = k^n + a_{k-1}k^{n-1} + a_{k-2}k^{n-2} + \cdots + a_1 k + a_0$ posiada następujące właściwości:

- jego stopień n jest równy liczbie wierzchołków grafu G,
- współczynnik liczbowy stojący przy najwyższej potędze k, czyli przy k^n, jest równy 1, wielomian o takiej właściwości nazywa się monicznym (ang. monic) lub unormowanym,
- wyraz wolny a_0 jest równy 0; Dowód: Liczba prawidłowych kolorowań wierzchołków grafu za pomocą $k = 0$ kolorów wynosi zero. Ponieważ $P_G(0) = a_0$, więc musi być $a_0 = 0$, w przeciwnym razie otrzymalibyśmy sprzeczność.
- współczynnik przy k^{n-1} jest równy z minusem liczbie krawędzi grafu, tzn. $a_{k-1} = -|E(G)|$; na wielomian chromatyczny można patrzeć jak na wzór wynikający z kombinatorycznej metody zliczania zwanej metodą włączania\wyłączania. Aby wyznaczyć liczbę kolorowań prawidłowych, należy od wszystkich kolorowań odjąć liczbę kolorowań nieprawidłowych. Ponieważ kolorujemy zaetykietowane wierzchołki grafu k kolorami, to wszystkich możliwych kolorowań będzie tyle, ile jest wariacji n-elementowych z powtórzeniami zbioru k-elementowego, czyli $W_k^n = k^n$ (n może być większe od k, równe k albo mniejsze od k). Od tej liczby k^n należy odjąć wszystkie te nieprawidłowe kolorowania, w których dokładnie jedna krawędź ma oba końce w tym samym kolorze. Mamy takich kolorowań tyle, ile jest krawędzi razy $W_k^{n-1} = k^{n-1}$, gdyż dwa wierzchołki połączone krawędzią muszą mieć ten sam kolor, a więc mamy jakby o jeden mniej wierzchołków do kolorowania. Ponieważ za dużo odjęliśmy (dlaczego?), to do wyniku należy teraz dodać liczbę nieprawidłowych kolorowań, w których dwie krawędzie (sąsiadujące lub nie) mają oba końce w tym samym kolorze (gdy sąsiadują) albo w tych samych dwóch kolorach (gdy nie sąsiadują), itd.
- zawsze wszystkie współczynniki $a_{k-1}, a_{k-2}, \ldots, a_2, a_1$ przyjmują niezerowe wartości,
- współczynniki wielomianu uporządkowanego według malejących potęg zmiennej k są na przemian dodatnie i ujemne(np. $k^{10}-13k^9+78k^8-282k^7+675k^6-1107k^5+1242k^4-918k^3+405k^2-81k$,
- jeżeli graf nie jest pusty, to suma współczynników jego wielomianu jest równa 0 (suma współczynników wielomianu grafu pustego N_n wynosi 1, bo ma on postać k^n). Dowód: Jedynymi grafami, które mają liczbę chromatyczną równą 1, są grafy puste N_n, zatem wszystkie pozostałe grafy mają liczbę chromatyczną równą co najmniej 2. Oznacza to, że dla tych grafów liczba prawidłowych kolorowań za pomocą co najwyżej 1 koloru ($k = 1$) wynosi zero, czyli $P_G(1) = 0$, ale $P_G(1) = 1^n + a_{k-1}1^{n-1} + a_{k-2}1^{n-2} + \cdots + a_1 1 + a_0 = 1 +$

$a_{k-1} + a_{k-2} + \cdots + a_1 + a_0 = 0$. Zapisana suma jest oczywiście sumą współczynników wielomianu chromatycznego, no i jest równa zeru, co należało udowodnić.

- wykładnik przy najniższej potędze k (z niezerowym współczynnikiem) jest równy liczbie składowych spójnych grafu; zatem wielomian chromatyczny każdego grafu spójnego „kończy się” składnikiem k^1, czyli k.
- ciąg współczynników wielomianu chromatycznego, uporządkowanego według malejących potęg zmiennej k, tworzy ciąg liczbowy logarytmicznie wklęsły (ang. logarithmically concave sequence), którego wyrazy, z wyjątkiem pierwszego i ostatniego, spełniają nierówność $a_i^2 \geq a_{i-1}a_{i+1}$. Oznacza to, że współczynniki wielomianu, co do wartości bezwzględnej, najpierw rosną, a następnie maleją.

Wyznaczyć liczbę chromatyczną za pomocą funkcji z biblioteki grafowej graphs:

```
(%i67) chromatic_number(graf);
(%o67) 3
```

Za pomocą funkcji vertex_coloring() z biblioteki graphs wyznaczyć jedno z wielu prawidłowych kolorowań wierzchołków za pomocą minimalnej liczby kolorów (równej liczbie chromatycznej). Jeżeli graf ma więcej wierzchołków niż wynosi jego liczba chromatyczna, to oczywiście niektóre wierzchołki (ale nie sąsiadujące ze sobą!) będą miały ten sam kolor. Z powodu przybliżonego algorytmu wyznaczania tego optymalnego kolorowania przez funkcję vertex_coloring(), może wyjść liczba chromatyczna inna niż z wielomianu! Oczywiście będzie to wartość błędna, gdyż graf ma tylko jedną liczbę chromatyczną.

```
(%i94) [liczba_chrom, kolorowanie]:vertex_coloring(graf)$
       liczba_chrom;
       kolorowanie;
       sort(kolorowanie);
       kolory_wierzch:makelist(para[2],para,sort(kolorowanie));
(%o91) 3
(%o92) [[4,3],[3,2],[2,1],[1,1],[0,3]]
(%o93) [[0,3],[1,1],[2,1],[3,2],[4,3]]
(kolory_wierzch) [3,1,1,2,3]
```

W bibliotece kombinatoryczno-grafowej znajduje się funkcja, która wyznacza wszystkie prawidłowe kolorowania wierzchołków za pomocą co najwyżej k kolorów i tych kolorowań jest tyle, ile wynosi wartość wielomianu chromatycznego dla zadanego k.

```
graph_all_proper_vertices_colorings_at_most(graf, k) - zwraca listę wszystkich
                                       prawidłowych kolorowań wierzchołków grafu graf za
                                       pomocą co najwyżej k kolorów.
```

Wygenerować wszystkie te kolorowania i **a)** policzyć, czy jest ich tyle, ile wynika z wielomianu chromatycznego oraz **b)** poszukać wśród wygenerowanych kolorowań tego, które zwróciła funkcja vertex_coloring() oraz te trzy, które należało narysować wcześniej. Funkcja … at_most() każde kolorowanie koduje w ten sposób, że najpierw podaje kolor wierzchołka nr 1, potem kolor wierzchołka nr 2 itd., dlatego konieczne było posortowanie kolorowania zwróconego przez funkcję vertex_coloring().

```
(%i95) graph_all_proper_vertices_colorings_at_most(p_graf, 3);
(%o95) [[1,2,1,3,1],[1,2,2,3,1],[1,3,1,2,1],[1,3,3,2,1],[2,1,1,3,2],[2,1,2,3,2],[2,3,2,1,2],
[2,3,3,1,2],[3,1,1,2,3],[3,1,3,2,3],[3,2,2,1,3],[3,2,3,1,3]]
```

Dla kilku otrzymanych kolorowań narysować graf i pokolorować jego wierzchołki, żeby przekonać się, że rzeczywiście kolorowania są prawidłowe.

Wygenerować wszystkie prawidłowe kolorowania za pomocą liczby kolorów o jeden większej od liczby chromatycznej i poszukać tego samego kolorowania na liście wynikowej. Zwrócić uwagę na to, że wśród kolorowań są także te kolorowania, w których nie użyto czterech kolorów, ale tylko 3. Te kolorowania będą używać trzech kolorów oznaczonych nie tylko liczbami 1,2,3, ale też np. liczbami 1,3,4; 2,3,4 itp. Napisać kilka instrukcji w Maximie, które z listy kolorowań at_most odfiltrują kolorowania za pomocą liczby kolorów równej liczbie chromatycznej (czyli tu za pomocą 3 kolorów wśród kolorowań za pomocą 4).

```
(%i101) Kolorowania:graph_all_proper_vertices_colorings_at_most(p_graf, 4);
        length(Kolorowania);
(Kolorowania) [[1,2,1,3,1],[1,2,1,3,4],[1,2,1,4,1],[1,2,1,4,3],[1,2,2,3,1],[1,2,2,3,4],[1,2,2,4,1],[
1,2,2,4,3],[1,2,3,4,1],[1,2,3,4,3],[1,2,4,3,1],[1,2,4,3,4],[1,3,1,2,1],[1,3,1,2,4],[1,3,1,4,1]
,[1,3,1,4,2],[1,3,2,4,1],[1,3,2,4,2],[1,3,3,2,1],[1,3,3,2,4],[1,3,3,4,1],[1,3,3,4,2],[1,3,4,2,
1],[1,3,4,2,4],[1,4,1,2,1],[1,4,1,2,3],[1,4,1,3,1],[1,4,1,3,2],[1,4,2,3,1],[1,4,2,3,2],[1,4,3,
2,1],[1,4,3,2,3],[1,4,4,2,1],[1,4,4,2,3],[1,4,4,3,1],[1,4,4,3,2],[2,1,1,3,2],[2,1,1,3,4],[2,1,
1,4,2],[2,1,1,4,3],[2,1,2,3,2],[2,1,2,3,4],[2,1,2,4,2],[2,1,2,4,3],[2,1,3,4,2],[2,1,3,4,3],[2,
1,4,3,2],[2,1,4,3,4],[2,3,1,4,1],[2,3,1,4,2],[2,3,2,1,2],[2,3,2,1,4],[2,3,2,4,1],[2,3,2,4,2],[
2,3,3,1,2],[2,3,3,1,4],[2,3,3,4,1],[2,3,3,4,2],[2,3,4,1,2],[2,3,4,1,4],[2,4,1,3,1],[2,4,1,3,2]
,[2,4,2,1,2],[2,4,2,1,3],[2,4,2,3,1],[2,4,2,3,2],[2,4,3,1,2],[2,4,3,1,3],[2,4,4,1,2],[2,4,4,1,
3],[2,4,4,3,1],[2,4,4,3,2],[3,1,1,2,3],[3,1,1,2,4],[3,1,1,4,2],[3,1,1,4,3],[3,1,2,4,2],[3,1,2,
4,3],[3,1,3,2,3],[3,1,3,2,4],[3,1,3,4,2],[3,1,3,4,3],[3,1,4,2,3],[3,1,4,2,4],[3,2,1,4,1],[3,2,
1,4,3],[3,2,2,1,3],[3,2,2,1,4],[3,2,2,4,1],[3,2,2,4,3],[3,2,3,1,3],[3,2,3,1,4],[3,2,3,4,1],[3,
2,3,4,3],[3,2,4,1,3],[3,2,4,1,4],[3,4,1,2,1],[3,4,1,2,3],[3,4,2,1,2],[3,4,2,1,3],[3,4,3,1,2],[
3,4,3,1,3],[3,4,3,2,1],[3,4,3,2,3],[3,4,4,1,2],[3,4,4,1,3],[3,4,4,2,1],[3,4,4,2,3],[4,1,1,2,3]
,[4,1,1,2,4],[4,1,1,3,2],[4,1,1,3,4],[4,1,2,3,2],[4,1,2,3,4],[4,1,3,2,3],[4,1,3,2,4],[4,1,4,2,
3],[4,1,4,2,4],[4,1,4,3,2],[4,1,4,3,4],[4,2,1,3,1],[4,2,1,3,4],[4,2,2,1,3],[4,2,2,1,4],[4,2,2,
3,1],[4,2,2,3,4],[4,2,3,1,3],[4,2,3,1,4],[4,2,4,1,3],[4,2,4,1,4],[4,2,4,3,1],[4,2,4,3,4],[4,3,
1,2,1],[4,3,1,2,4],[4,3,2,1,2],[4,3,2,1,4],[4,3,3,1,2],[4,3,3,1,4],[4,3,3,2,1],[4,3,3,2,4],[4,
3,4,1,2],[4,3,4,1,4],[4,3,4,2,1],[4,3,4,2,4]]
(%o101) 144
```

Użyć funkcji …exact(), aby wygenerować tylko te kolorowania, w których użyto dokładnie tylu kolorów, ile zadano, tzn. każdy z kolorów został użyty co najmniej raz (do pokolorowania co najmniej jednego wierzchołka).

```
(%i103) Kolorowania:graph_all_proper_vertices_colorings_exact(p_graf, 4);
        length(Kolorowania);
(Kolorowania) [[1,2,4,3,4],[1,3,4,2,4],[1,4,4,2,3],[1,4,4,3,2],[2,1,4,3,4],[2,3,4,1,4],[2,4,4,1,3],[
2,4,4,3,1],[3,1,4,2,4],[3,2,4,1,4],[3,4,4,1,2],[3,4,4,2,1],[4,1,2,3,4],[4,1,3,2,4],[4,1,4,2,3]
,[4,1,4,3,2],[4,2,1,3,4],[4,2,3,1,4],[4,2,4,1,3],[4,2,4,3,1],[4,3,1,2,4],[4,3,2,1,4],[4,3,4,1,
2],[4,3,4,2,1],[1,2,3,4,3],[1,3,3,2,4],[1,3,3,4,2],[1,4,3,2,3],[2,1,3,4,3],[2,3,3,1,4],[2,3,3,
4,1],[2,4,3,1,3],[3,1,2,4,3],[3,1,3,2,4],[3,1,3,4,2],[3,1,4,2,3],[3,2,1,4,3],[3,2,3,1,4],[3,2,
3,4,1],[3,2,4,1,3],[3,4,1,2,3],[3,4,2,1,3],[3,4,3,1,2],[3,4,3,2,1],[4,1,3,2,3],[4,2,3,1,3],[4,
3,3,1,2],[4,3,3,2,1],[1,2,2,3,4],[1,2,2,4,3],[1,3,2,4,2],[1,4,2,3,2],[2,1,2,3,4],[2,1,2,4,3],[
2,1,3,4,2],[2,1,4,3,2],[2,3,1,4,2],[2,3,2,1,4],[2,3,2,4,1],[2,3,4,1,2],[2,4,1,3,2],[2,4,2,1,3]
,[2,4,2,3,1],[2,4,3,1,2],[3,1,2,4,2],[3,2,2,1,4],[3,2,2,4,1],[3,4,2,1,2],[4,1,2,3,2],[4,2,2,1,
3],[4,2,2,3,1],[4,3,2,1,2],[1,2,1,3,4],[1,2,1,4,3],[1,2,3,4,1],[1,2,4,3,1],[1,3,1,2,4],[1,3,1,
4,2],[1,3,2,4,1],[1,3,4,2,1],[1,4,1,2,3],[1,4,1,3,2],[1,4,2,3,1],[1,4,3,2,1],[2,1,1,3,4],[2,1,
1,4,3],[2,3,1,4,1],[2,4,1,3,1],[3,1,1,2,4],[3,1,1,4,2],[3,2,1,4,1],[3,4,1,2,1],[4,1,1,2,3],[4,
1,1,3,2],[4,2,1,3,1],[4,3,1,2,1]]
(%o103) 96
```

Z wielomianu chromatycznego nie da się bezpośrednio obliczyć liczby kolorowań typu exact, należy zastosować wzory podane niżej. Aby obliczyć Cdok(k), zamiast C(k) należy użyć funkcji P(k), czyli wielomianu chromatycznego. Wykorzystując podane wzory, wyznaczyć liczbę kolorowań typu exact dla liczby kolorów o jeden większej od liczby chromatycznej i przekonać się, że ta liczba kolorowań jest równa tyle, ile kolorowań zwróciła funkcja …exact().

```
Cdok(k) := sum( k! / ( (k+1-i)!*(i-1)! ) * (-1)^(i+1) * C(k+1-i),  i, 1, k );

Cdok( 1) = C( 1)
Cdok( 2) = C( 2) -  2*C( 1)
Cdok( 3) = C( 3) -  3*C( 2) +  3*C( 1)
Cdok( 4) = C( 4) -  4*C( 3) +  6*C( 2) -  4*C( 1)
Cdok( 5) = C( 5) -  5*C( 4) + 10*C( 3) - 10*C( 2) +   5*C( 1)
Cdok( 6) = C( 6) -  6*C( 5) + 15*C( 4) - 20*C( 3) +  15*C( 2) -   6*C( 1)
Cdok( 7) = C( 7) -  7*C( 6) + 21*C( 5) - 35*C( 4) +  35*C( 3) -  21*C( 2) +   7*C( 1)
Cdok( 8) = C( 8) -  8*C( 7) + 28*C( 6) - 56*C( 5) +  70*C( 4) -  56*C( 3) +  28*C( 2) -   8*C( 1)
Cdok( 9) = C( 9) -  9*C( 8) + 36*C( 7) - 84*C( 6) + 126*C( 5) - 126*C( 4) +  84*C( 3) -  36*C( 2) +   9*C( 1)
Cdok(10) = C(10) - 10*C( 9) + 45*C( 8) -120*C( 7) + 210*C( 6) - 252*C( 5) + 210*C( 4) - 120*C( 3) +  45*C( 2) -  10*C( 1)
```

Funkcja …exact2() pozwala wygenerować jeszcze bardziej precyzyjnie określone kolorowania, tzn. kolorowania wybranym zestawem kolorów, w którym podajemy, ile razy dany kolor ma być użyty. Kolorowanie [3,1,1,2,3] jest kolorowaniem typu [2,1,2], w którym koloru 1. użyto dwa razy, koloru 2. użyto raz i koloru 3. użyto 2 razy. 2+1+2=5, ponieważ kolorujemy 5 wierzchołków. Wygenerować wszystkie kolorowania takiego typu, jakie zwróciła funkcja vertex_coloring() i odszukać tego kolorowania na liście. Dlaczego jest tak mało kolorowań typu exact2? Z czego to wynika?

```
(%i110)   Kolorowania:graph_all_proper_vertices_colorings_exact2(p_graf, [2,1,2]);
          length(Kolorowania);
(Kolorowania) [[1,3,3,2,1],[3,1,1,2,3]]
(%o110)   2
```

Jeżeli interesuje nas tylko liczba kolorowań, bez ich generowania, to można użyć funkcji przeznaczonych do zliczania kolorowań, które oczywiście działają szybciej:

```
(%i104)   number_of_proper_vertices_colorings_at_most(p_graf,4);
(%o104)   144

(%i107)   number_of_proper_vertices_colorings_exact(p_graf,4);
(%o107)   96

(%i106)   number_of_proper_vertices_colorings_exact2(p_graf,[2,1,2]);
(%o106)   2
```

Jeżeli chcemy uzyskać zbiorczą informację o liczbach kolorowań różnymi zestawami co najwyżej np. czterech kolorów albo dokładnie czterema kolorami, wówczas należy użyć procedury generującej tzw. funkcję tworzącą ciągu liczb kolorowań, czyli wielomian wielu zmiennych, w którym współczynniki jednomianów oznaczają liczby kolorowań, a wykładnik każdej zmiennej mówi, ile razy został użyty dany kolor. Kolory są tu nazwane literami (symbolami), gdyż liczby mogłyby być źle zinterpretowane.

Suma wszystkich współczynników wielomianu funkcji tworzącej … at_most() jest równa liczbie kolorowań za pomocą co najwyżej zadanej liczby kolorów, natomiast suma wszystkich współczynników wielomianu funkcji tworzącej … exact() jest równa liczbie kolorowań za pomocą dokładnie zadanej liczby kolorów.

```
(%i132) funkcja_tworzaca:graph_gen_fun_for_proper_vertices_colorings_at_most(p_graf,[r,g,b,y]);
```

(funkcja_tworzaca) $$2\,g\,r\,y^3+2\,b\,r\,y^3+2\,b\,g\,y^3+2\,g\,r^2\,y^2+2\,b\,r^2\,y^2+2\,g^2\,r\,y^2+24\,b\,g\,r\,y^2+2\,b^2\,r\,y^2+2\,b\,g^2\,y^2+2\,b^2\,g\,y^2+2\,g\,r^3\,y+2\,b\,r^3\,y+2\,g^2\,r^2\,y+24\,b\,g\,r^2\,y+2\,b^2\,r^2\,y+2\,g^3\,r\,y+24\,b\,g^2\,r\,y+24\,b^2\,g\,r\,y+2\,b^3\,r\,y+2\,b\,g^3\,y+2\,b^2\,g^2\,y+2\,b^3\,g\,y+2\,b\,g\,r^3+2\,b\,g^2\,r^2+2\,b^2\,g\,r^2+2\,b\,g^3\,r+2\,b^2\,g^2\,r+2\,b^3\,g\,r$$

```
(%i134) wspolczynniki:subst([r=1,g=1,b=1,y=1],args(funkcja_tworzaca));
        sum(wspolczynniki[i],i,1,length(wspolczynniki));
(wspolczynniki) [2,2,2,2,2,2,24,2,2,2,2,2,2,24,2,2,24,24,2,2,2,2,2,2,2,2,2,2]
(%o134) 144
```

```
(%i135) Funkcja_tworzaca:graph_gen_fun_for_proper_vertices_colorings_exact(p_graf,[r,g,b,y]);
```

(Funkcja_tworzaca) $$24\,b\,g\,r\,y^2+24\,b\,g\,r^2\,y+24\,b\,g^2\,r\,y+24\,b^2\,g\,r\,y$$

```
(%i137) wspolczynniki:subst([r=1,g=1,b=1,y=1],args(Funkcja_tworzaca));
        sum(wspolczynniki[i],i,1,length(wspolczynniki));
(wspolczynniki) [24,24,24,24]
(%o137) 96
```

Sprawdzić, za pomocą funkcji …exact2(), dla wybranego składnika obu funkcji tworzących, czy współczynnik liczbowy stojący przy wybranym jednomianie ma poprawną wartość.

Wyznaczyć, korzystając z odpowiedniego wzoru na Cdok(k), liczbę prawidłowych kolorowań wierzchołków grafu za pomocą dokładnie tylu kolorów, ile wynosi n (liczba wierzchołków grafu). Przekonać się, że ta liczba kolorowań wynosi n!. Dlaczego?

Wyznaczyć, korzystając z odpowiedniego wzoru na Cdok(k), liczbę prawidłowych kolorowań wierzchołków grafu za pomocą dokładnie tylu kolorów, ile wynosi n+1 (liczba wierzchołków grafu + 1). Przekonać się, że ta liczba kolorowań wynosi 0. Dlaczego? Czy otrzymamy zero także dla n+2?

Dla ambitnych. Jeżeli przyjrzeć się dowolnemu grafowi, którego wierzchołki zostały pokolorowane prawidłowo, to łatwo zauważyć, że wierzchołki pokolorowane jednym i tym samym kolorem stanowią zbiór niezależny. Kolorowanie wierzchołków prowadzi więc w istocie do podziału wierzchołków grafu na zbiory niezależne. Liczba chromatyczna grafu mówi zatem, jaka musi być minimalna liczba zbiorów niezależnych, aby każdy wierzchołek grafu znalazł się w jednym z nich. Stosując algebrę Boole'a, wyznaczyć liczbę chromatyczną badanego grafu (przybliżoną, która może wyjść taka, jak z wielomianu chromatycznego) w następujący sposób: Wyznaczyć największy maksymalny zbiór niezależny grafu *G* i wszystkie wierzchołki tego zbioru niezależnego pokolorować tym samym kolorem nr 1. Usunąć te wierzchołki z grafu i w nowym grafie znowu wyznaczyć największy zbiór niezależny, pokolorować te wierzchołki kolorem nr 2, i znowu usunąć je z grafu. Kontynować, aż powstanie graf pusty. Liczba użytych kolorów w tym algorytmie będzie w przybliżeniu (albo dokładnie) równa liczbie chromatycznej grafu.

2. Planowanie zajęć za pomocą kolorowania wierzchołków grafu konfliktów zajęć

Oznaczmy przez *Zi*(*Gj*, *Nk*, *Sl*) zajęcia, w których uczestniczy grupa *Gj*, prowadzone przez nauczyciela *Nk*, odbywające się w sali *Sl*. Rozplanować zajęcia dla grup laboratoryjnych *G*1, *G*2, *G*3, *G*4, *G*5, prowadzone przez nauczycieli *N*1, *N*2, w salach *S*1, *S*2, *S*3 tak, aby wszystkie zajęcia odbyły się w możliwie najkrótszym czasie, tzn. w najmniejszej liczbie terminów (jeden termin to dwie godziny zajęć). Sala *S*1 to sala laboratoryjna, sala *S*2 to sala ćwiczeniowa, a sala *S*3 to sala wykładowa. Zajęcia do rozplanowania: *Z*1(*G*1, *N*1, *S*1), *Z*2(*G*2, *N*1, *S*1), *Z*3(*G*3, *N*2, *S*1), *Z*4(*G*4, *N*1, *S*1), *Z*5(*G*5, *N*2, *S*1), *Z*6(*G*1 + *G*2, *N*2, *S*2), *Z*7(*G*3 + *G*4, *N*1, *S*2), *Z*8(*G*5, *N*1, *S*2), *Z*9(*G*1 + *G*2 + *G*5, *N*2, *S*3), *Z*10(*G*3 + *G*4, *N*2, *S*3). Grupy powiązane znakiem „+” odbywają zajęcia razem, w tej samej sali, z tym samym nauczycielem. W celu rozwiązania tego problemu należy narysować graf, którego wierzchołkami są zajęcia *Z*1, *Z*2,…, *Z*10. Wierzchołki *Zi* oraz *Zj* tego grafu są połączone krawędzią wtedy i tylko wtedy, gdy w zajęciach *Zi* oraz *Zj* uczestniczą te same grupy studentów lub prowadzą je ci sami nauczyciele, lub odbywają się w tych samych salach. Niektóre krawędzie powtórzą się kilka razy, ale takie multikrawędzie należy zastąpić pojedynczymi krawędziami.

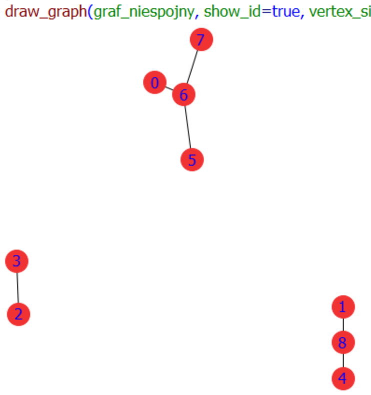
Minimalna liczba kolorów potrzebna do prawidłowego pokolorowania wierzchołków utworzonego grafu będzie równa minimalnej liczbie różnych terminów potrzebnych do bezkolizyjnego przeprowadzenia zajęć. Zajęcia pokolorowane tym samym kolorem mogą odbywać się w tym samym czasie. „Ręczne” szukanie wielomianu chromatycznego i liczby chromatycznej jest zbyt uciążliwe w tym dosyć dużym grafie, dlatego zadanie prawidłowego kolorowania jego wierzchołków najlepiej rozwiązać w programie Maxima. Wyznaczenie liczby chromatycznej nie kończy zadania, bo informuje tylko o tym, ile co najmniej potrzeba terminów do bezkolizyjnego przeprowadzenia zajęć. Aby przypisać konkretne zajęcia do konkretnego terminu, należy użyć odpowiedniej funkcji generującej kolorowania. Zadanie będzie miało wiele rozwiązań. Dla jednego z nich zbudować tabele planu zajęć, które w wierszach będą miały terminy, a w kolumnach grupy laboratoryjne; w wierszach będą terminy, a w kolumnach nazwiska nauczycieli; w wierszach będą terminy, a w kolumnach nazwy wszystkich dostępnych sal.

3. Wyznaczanie wielomianu chromatycznego grafu niespójnego

Przepisać poniższe instrukcje do Maximy i wylosować graf niespójny. Wyznaczyć wielomian chromatyczny wylosowanego grafu korzystając z faktu, że wielomian chromatyczny grafu niespójnego jest równy iloczynowi wielomianów każdej składowej spójnej, tzn. $P_G(k) = P_{G1}(k)P_{G2}(k)\dots P_{Gr}(k)$, gdzie r to liczba składowych spójności, a $P_{G1}(k)$, $P_{G2}(k), \dots, P_{Gr}(k)$ to wielomiany chromatyczne tych komponent. Skorzystać ze wzorów na wielomian drzewa, cyklu lub grafu pełnego, a w razie konieczności usuwać lub dodawać krawędzie, żeby wyznaczyć wielomian każdej składowej spójnej. Sprawdzić funkcją chromatic_polynomial(), czy iloczyn wielomianów jest taki sam, jak wyznaczony wielomian całego grafu niespójnego. Mnożenie wielomianów i inne obliczenia wykonać w Maximie.

```
(%i31) do (
         graf_niespojny : random_graph(9,0.4),
         if not is_connected(graf_niespojny) then
         (
           skladowe_spojne:connected_components(graf_niespojny),
           lista:makelist(length(s),s, skladowe_spojne),
           if every(lambda([s], s>=2),lista) and length(skladowe_spojne)>=3 then  return(0)
         )
       )$
       draw_graph(graf_niespojny, show_id=true, vertex_size=4)$
```

(%t31)

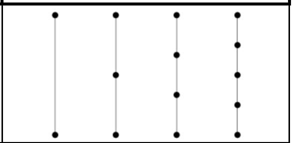
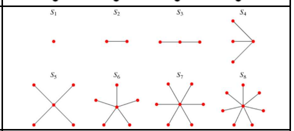
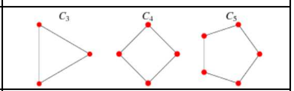
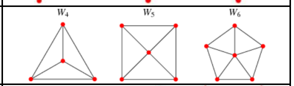
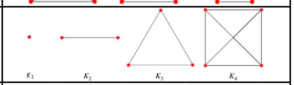
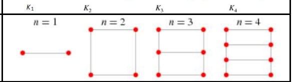
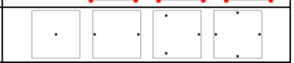
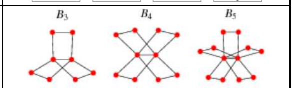


4. Prawidłowe kolorowanie krawędzi grafu

Minimalną liczbę kolorów potrzebną do prawidłowego pokolorowania krawędzi grafu nazywamy krawędziową liczbą chromatyczną lub indeksem chromatycznym grafu. Okazuje się (tw. Vizinga https://en.wikipedia.org/wiki/Vizing%27s_theorem), że indeks chromatyczny każdego grafu jest równy albo najwyższemu stopniowi wierzchołka w tym grafie, albo jest o 1 od niego większy. Aby wyznaczyć indeks chromatyczny grafu, można utworzyć graf krawędziowy (ang. Line graph) dla danego grafu i kolorować wierzchołki grafu krawędziowego, dlatego nie istnieje w teorii grafów takie pojęcie, jak wielomian chromatyczny dla krawędzi.

Dla grafu G wylosowanego w punkcie 1. wyznaczyć na kartce papieru graf krawędziowy $L(G)$. Wprowadzić graf $L(G)$ do Maximy za pomocą funkcji create_graph(). Wyznaczyć wielomian chromatyczny grafu krawędziowego $L(G)$, a następnie jego liczbę chromatyczną. Liczba ta będzie równocześnie indeksem chromatycznym oryginalnego grafu G. Funkcją max_degree() wyznaczyć największy stopień wierzchołka w grafie oryginalnym i porównać z otrzymanym indeksem chromatycznym. W Maximie użyć funkcji chromatic_index() do bezpośredniego wyznaczenia indeksu chromatycznego oryginalnego grafu. Natomiast funkcją edge_coloring() wyznaczyć jedno z wielu prawidłowych kolorowań krawędzi grafu za pomocą minimalnej liczby kolorów. Z biblioteki kombinatoryczno-grafowej użyć odpowiednich funkcji do generowania kolorowań krawędzi oraz do zliczania kolorowań krawędzi i przeprowadzić podobne badanie, jak w punkcie pierwszym tej instrukcji, gdzie badano kolorowania wierzchołków grafu.

Wielomiany chromatyczne popularnych rodzin grafów:

Graf	Wielomian chromatyczny	Uwaga	Wygląd grafu
ścieżkowy P_n	$k(k-1)^{n-1}$	$n \geq 1$	
gwiazdowy S_n	$k(k-1)^{n-1}$	$n \geq 1$	
dowolne drzewo n-wierzchołkowe T_n	$k(k-1)^{n-1}$	$n \geq 1$	
las n-wierzchołkowy, m-krawędziowy, c-komponentowy	$(-1)^{n-c}k^c(1-k)^m$	$n \geq c$	
cykliczny C_n	$(k-1)^n + (-1)^n(k-1)$	$n \geq 3$	
kołowy W_n	$k(k-2)^{n-1} - (-1)^n k(k-2)$	$n \geq 4$	
pełny K_n	$k(k-1)(k-2)\dots(k-(n-1))$	$n \geq 1$	
drabinkowy $P_2 \times P_n$	$k(k-1)(k^2-3k+3)^{n-1}$	$n \geq 1$	
pusty N_n	k^n	$n \geq 1$	
Książkowy $B_n = S_{n+1} \times P_2$	$k(k-1)(k^2-3k+3)^n$	$n \geq 3$	

Wielomian chromatyczny, w postaci rozwiniętej, dla grafów P_n, S_n i drzew T_n:

```
print_list(makelist(expand(k·(k−1)^(n−1)), n, 1,10))$
```

k
k^2-k
k^3-2k^2+k
$k^4-3k^3+3k^2-k$
$k^5-4k^4+6k^3-4k^2+k$
$k^6-5k^5+10k^4-10k^3+5k^2-k$
$k^7-6k^6+15k^5-20k^4+15k^3-6k^2+k$
$k^8-7k^7+21k^6-35k^5+35k^4-21k^3+7k^2-k$
$k^9-8k^8+28k^7-56k^6+70k^5-56k^4+28k^3-8k^2+k$
$k^{10}-9k^9+36k^8-84k^7+126k^6-126k^5+84k^4-36k^3+9k^2-k$

Wielomian chromatyczny, w postaci rozwiniętej, dla grafów cyklicznych C_n:

```
print_list(makelist(expand( (k−1)^(n) + (−1)^n·(k−1) ), n, 1,10))$
```

0
k^2-k
k^3-3k^2+2k
$k^4-4k^3+6k^2-3k$
$k^5-5k^4+10k^3-10k^2+4k$
$k^6-6k^5+15k^4-20k^3+15k^2-5k$
$k^7-7k^6+21k^5-35k^4+35k^3-21k^2+6k$
$k^8-8k^7+28k^6-56k^5+70k^4-56k^3+28k^2-7k$
$k^9-9k^8+36k^7-84k^6+126k^5-126k^4+84k^3-36k^2+8k$
$k^{10}-10k^9+45k^8-120k^7+210k^6-252k^5+210k^4-120k^3+45k^2-9k$

Wielomian chromatyczny, w postaci iloczynowej, dla grafów cyklicznych C_n:

```
print_list(makelist(factor( (k−1)^(n) + (−1)^n·(k−1) ), n, 1,10))$
```

0
$(k-1)\,k$
$(k-2)\,(k-1)\,k$
$(k-1)\,k\,(k^2-3k+3)$
$(k-2)\,(k-1)\,k\,(k^2-2k+2)$
$(k-1)\,k\,(k^4-5k^3+10k^2-10k+5)$
$(k-2)\,(k-1)\,k\,(k^2-3k+3)\,(k^2-k+1)$
$(k-1)\,k\,(k^6-7k^5+21k^4-35k^3+35k^2-21k+7)$
$(k-2)\,(k-1)\,k\,(k^2-2k+2)\,(k^4-4k^3+6k^2-4k+2)$
$(k-1)\,k\,(k^2-3k+3)\,(k^6-6k^5+15k^4-21k^3+18k^2-9k+3)$

Wielomian chromatyczny, w postaci rozwiniętej, dla grafów kołowych W_n:

```
print_list(makelist(expand( k·(k−2)^(n−1) − (−1)^n·k·(k−2) ), n, 4,10))$
```

$k^4-6k^3+11k^2-6k$
$k^5-8k^4+24k^3-31k^2+14k$
$k^6-10k^5+40k^4-80k^3+79k^2-30k$
$k^7-12k^6+60k^5-160k^4+240k^3-191k^2+62k$
$k^8-14k^7+84k^6-280k^5+560k^4-672k^3+447k^2-126k$
$k^9-16k^8+112k^7-448k^6+1120k^5-1792k^4+1792k^3-1023k^2+254k$
$k^{10}-18k^9+144k^8-672k^7+2016k^6-4032k^5+5376k^4-4608k^3+2303k^2-510k$

Wielomian chromatyczny, w postaci iloczynowej, dla grafów kołowych W_n:

```
print_list(makelist(factor( k·(k−2)^(n−1) − (−1)^n·k·(k−2) ), n, 4,10))$
```

$(k-3)(k-2)(k-1)k$
$(k-2)(k-1)k(k^2-5k+7)$
$(k-3)(k-2)(k-1)k(k^2-4k+5)$
$(k-2)(k-1)k(k^4-9k^3+31k^2-49k+31)$
$(k-3)(k-2)(k-1)k(k^2-5k+7)(k^2-3k+3)$
$(k-2)(k-1)k(k^6-13k^5+71k^4-209k^3+351k^2-321k+127)$
$(k-3)(k-2)(k-1)k(k^2-4k+5)(k^4-8k^3+24k^2-32k+17)$

Wielomian chromatyczny, w postaci rozwiniętej, dla grafów drabinkowych $P_2 \times P_n$:

```
print_list( makelist( expand( k·(k−1)·(k^2−3·k+3)^(n−1) ), n, 1, 6) )$
```

k^2-k
$k^4-4k^3+6k^2-3k$
$k^6-7k^5+21k^4-33k^3+27k^2-9k$
$k^8-10k^7+45k^6-117k^5+189k^4-189k^3+108k^2-27k$
$k^{10}-13k^9+78k^8-282k^7+675k^6-1107k^5+1242k^4-918k^3+405k^2-81k$
$k^{12}-16k^{11}+120k^{10}-555k^9+1755k^8-3978k^7+6588k^6-7965k^5+6885k^4-4050k^3+1458k^2-243k$

Wielomian chromatyczny, w postaci iloczynowej, dla grafów pełnych K_n:

```
print_list( makelist( apply("·",makelist(k−(N−1),N,1,n)), n, 1, 10 ) )$
```

k
$(k-1)k$
$(k-2)(k-1)k$
$(k-3)(k-2)(k-1)k$
$(k-4)(k-3)(k-2)(k-1)k$
$(k-5)(k-4)(k-3)(k-2)(k-1)k$
$(k-6)(k-5)(k-4)(k-3)(k-2)(k-1)k$
$(k-7)(k-6)(k-5)(k-4)(k-3)(k-2)(k-1)k$
$(k-8)(k-7)(k-6)(k-5)(k-4)(k-3)(k-2)(k-1)k$
$(k-9)(k-8)(k-7)(k-6)(k-5)(k-4)(k-3)(k-2)(k-1)k$

Wielomian chromatyczny, w postaci rozwiniętej, dla grafów pełnych K_n:

```
print_list( makelist( expand(apply("·",makelist(k−(N−1),N,1,n))), n, 1, 10 ) )$
```

k

$k^2 - k$

$k^3 - 3k^2 + 2k$

$k^4 - 6k^3 + 11k^2 - 6k$

$k^5 - 10k^4 + 35k^3 - 50k^2 + 24k$

$k^6 - 15k^5 + 85k^4 - 225k^3 + 274k^2 - 120k$

$k^7 - 21k^6 + 175k^5 - 735k^4 + 1624k^3 - 1764k^2 + 720k$

$k^8 - 28k^7 + 322k^6 - 1960k^5 + 6769k^4 - 13132k^3 + 13068k^2 - 5040k$

$k^9 - 36k^8 + 546k^7 - 4536k^6 + 22449k^5 - 67284k^4 + 118124k^3 - 109584k^2 + 40320k$

$k^{10} - 45k^9 + 870k^8 - 9450k^7 + 63273k^6 - 269325k^5 + 723680k^4 - 1172700k^3 + 1026576k^2 - 362880k$

Wielomian chromatyczny nie może służyć jako matematyczne narzędzie rozstrzygające o izomorfizmie grafów, tzn. dwa nieizomorficzne grafy mogą posiadać taki sam wielomian chromatyczny. Nieizomorficzne grafy posiadające taki sam wielomian są nazywane chromatycznie równoważnymi. Natomiast graf, który jako jedyny posiada pewien wielomian chromatyczny, nazywany jest grafem chromatycznie unikalnym. Liczba chromatycznie unikalnych grafów prostych n-wierzchołkowych dla n od 1 do 10 wynosi: 1, 2, 4, 7, 16, 41, 139, 704, 7270, 183606 (https://oeis.org/A137568).