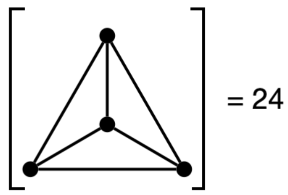
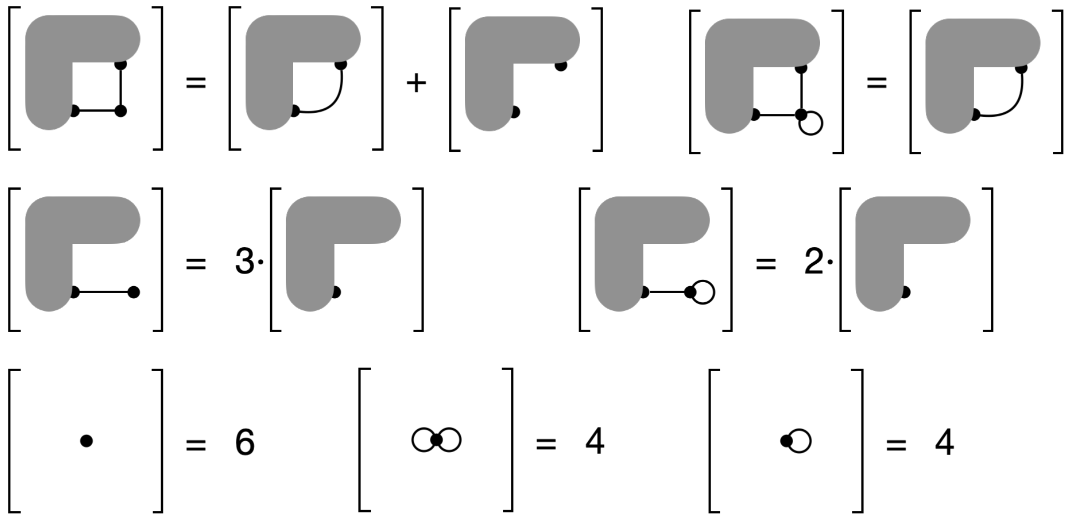
Projekt 2 [*] (Modele lodu)

Ważną klasą modeli w fizyce statystycznej są tzw. modele lodu (nazywane też modelami 6-wierzchołkowymi). Dany jest graf nieskierowany (zbiór wierzchołków i krawędzi łączących niektóre z par wierzchołków) którego każdy z wierzchołków ma co najwyżej 4 krawędzie. Krawędziom tego grafu przyporządkowujemy orientację (strzałki) tak aby spełniony był warunek: z każdego wierzchołka co najwyżej dwie strzałki wychodzą i do każdego wierzchołka co najwyżej dwie strzałki wchodzą. Oznaczmy przez nawias prostokątny liczbę konfiguracji strzałek dla danego grafu, np. (warto sprawdzić „ręcznie”):



Dla zadanego grafu problem zliczania konfiguracji strzałek można często uprościć korzystając z następujących reguł:



Napisz program który wczyta opis grafu wejściowego w postaci listy par wierzchołków (każda para to łącząca te wierzchołki krawędź), wielokrotnie użyje powyższych reguł i zwróci listę grafów których już nie da się dalej uprościć, wraz ze współczynnikami. Wskazówki: warto prowadzić obliczenia tak aby na każdym etapie liczba grafów była jak najmniejsza, a aby to zrobić warto utożsamiać grafy izomorficzne („takie same”, różniące się permutacją wierzchołków). Sprawdzanie czy dwa grafy są izomorficzne jest w ogólności trudnym problemem, można użyć biblioteki NetworkX (i funkcji `is_isomorphic`).

Przykład 1:
— input:
`[[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 1]]`

— output (lista par: `[liczebność, graf]`) gdzie graf nie daje się dalej uprościć używać reguł podanych w zadaniu:
`[[730, []]]`

Przykład 2:

```
input:
[[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 1], [1, 7], [3, 8], [5, 9],
[7, 10], [8, 10], [9, 10]]

output:
[[1, [[1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]]], [14826, []]]
```