

LABORATORIUM V

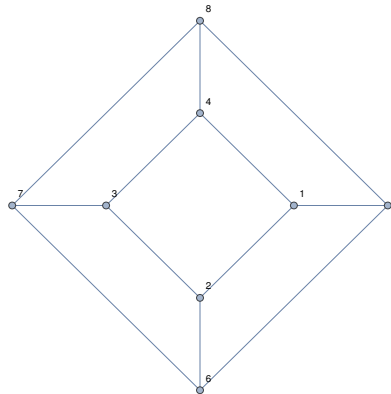
Przypomnijmy na początek znaczenie operacji wykorzystywanych w algorytmach, które będziemy dziś implementować.

- **top()** – pobranie elementu ze (szczytu) stosu.
- **pop()** – usunięcie elementu ze szczytu stosu.
- **push()** – włożenie elementu na stos.
- **enqueue()** – wstawienie elementu do kolejki.
- **dequeue()** – pobranie elementu z kolejki wraz z jego usunięciem.

ZADANIA

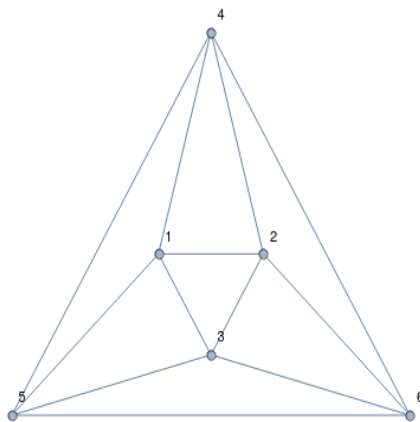
Zadanie 1 (obowiązkowe), 1 pkt. Uzupełnić kod w pliku `bipartite.cpp` tak, aby całość dawała odpowiedź na pytanie czy graf zadany w postaci list sąsiedztwa jest dwudzielny, czy nie. Program powinien być napisany w taki sposób, aby w przypadku pozytywnej odpowiedzi w tablicy `label` znajdowały się liczby 1 oraz 2 oznaczające do której klasy podziału należy dany wierzchołek.

Graf konstruowany w programie przedstawiony jest poniżej: ^(a):



^(a)Rysunek: Pakiet Mathematica (j.w.)

Zadanie 2 (obowiązkowe), 1 pkt. Uzupełnić kod w pliku `euler.cpp` tak, aby całość wypisywała na standardowe wyjście obwód Eulera grafu zadanego w postaci list sąsiedztwa. W rozwiązaniu należy wykorzystać podejście iteracyjne; dodatkowo graf konstruowany w kodzie przedstawiony jest poniżej:



(Jest to tzw. *Octahedral graph*; źródło: Pakiet Mathematica ^(b)).

Zadanie 3, 2 pkt. Napisać program, który wypisze na standardowe wyjście rozkład krawędzi podanego grafu eulrowskiego na rozłączne cykle (jak w twierdzeniu Eulera - Hierholzera). Dla grafu z poprzedniego zadania program mógłby wypisać na przykład

```
2 3 1 4 2
1 2 6 3 5 1
5 4 6 5
```

Dla testów w pliku `disjoint_cycles.cpp` "wpisany" jest graf z zadania 9. z zestawu 11. W rozwiązaniu należy przyjąć, iż graf będzie podany w postaci list sąsiedztwa.

^(b)Wolfram Research, Inc., Mathematica, Version 11.3, Champaign, IL (2018).