Sprawozdanie 1

Zadanie 1

1. Graf - zbiór wierzchołków połączonych krawędziami. Istnieją różne typy grafów np. graf skierowany, graf z wagami.
2. W grafie skierowanym krawędzie mają swój zwrot, zatem może nie istnieć przejście powrotne z jednego wierzchołka do drugiego, w grafie nieskierowanym dana krawędź umożliwia przejście w obie strony.
3. W grafie ważonym wierzchołki lub częściej krawędzie mają przypisane wagi, które mogą mieć różną interpretacje, np. w kontekście problemu komiwojażera wagi krawędzi oznaczają koszt przemieszczenia się z jednego wierzchołka do drugiego.
4. Graf spójny to taki graf, w którym dla każdych dwóch wierzchołków da się wyznaczyć ścieżkę, która je łączy.
5. Cykl to ścieżka po grafie, która zaczyna się i kończy w tym samym wierzchołku. Natomiast graf acykliczny to taki, w którym nie występują cykle.

Zadanie 2

W języku Python istnieje kilka bibliotek poświęconych zagadnieniu grafów m. in. py_graph, igraph, NetworkX, przydatna okazać się może również biblioteka numpy w przypadku przedstawiania grafu w postaci macierzy sąsiedztwa. Często polecanym rozwiązaniem w przypadku reprezentacji grafów w Pythonie jest użycie tzw. listy sąsiedztwa wykorzystując słowniki, które w prosty sposób pozwalają na zapisanie zależności między wierzchołkami. Wtedy do sprawdzenia istnienia krawędzi należy sprawdzić czy pod podanym kluczem znajduję się szukana wartość. Minusem takiego rozwiązania jest brak możliwości łatwego przypisania wag. Innym rozwiązaniem jest użycie macierzy sąsiedztwa. W tym celu można wykorzystać wbudowaną w bibliotekę numpy klasę matrix lub skorzystać z listy list. W tym przypadku już w prosty sposób można zapisać wagi poszczególnych krawędzi, do reprezentacji wag zerowych czy braku połączeń można wykorzystać stałe np. numpy.inf lub NaN. Wyszukiwanie elementów maksymalnych oraz minimalnych można przeprowadzić za pomocą wbudowanych funkcji min() oraz max().