**Program do wyszukiwania połączeń kolejowych**

Bartłomiej Dec

Grupa 22, lab 3

*1. Założenia projektu:*

Program miał na celu wyszukiwanie możliwych połączeń kolejowych między miastami. Użytkownik powinien mieć możliwość wyszukiwania spośród aktualnych połączeń, a także dodawać swoje aktualizacje m.in. dodanie połączenia bezpośredniego, usunięcia połączenia itp. Wyszukiwanie powinno odbywać sie zgodnie z algorytmem BFS(Breadth First Search) dla dwóch reprezentacji grafu (macierz sąsiedztwa lub lista sąsiedztwa). Punktem wyjściowym powinien być wspólny interfejs z zestawem funkcji do zaimplementowania. Do stworzenia graficznego interfejsu użyto biblioteki Tkinter.

*2. Graphical User Interface*

Panel składa się z dwóch okienek. Pierwsze z nich to okno modyfikacji, w którym możemy wprowadzić nowe połączenia lub usunąć aktualne. Drugie okno to wybór połączenia spośród dostępnych miast na dwóch rozwijanych listach.

*3. Skład projektu*

a)plik tworzący Interface Class, zestaw metod do zaimplementowania:

AbstractGraphInterface.py

b)plik z graficznym interfejsem użytkownika:

GUI.py

c)plik z implementacją Interface:

AdjacencyListClass.py

d)plik z przykładową siecią połączeń kolejowych, bazującą na rzeczywistym stanie:

SampleGraph.py

*4. Napotkane problemy*

Nie udało się podzielić implementacji algorytmu na dwie osobne możliwe reprezentacje grafu. Algorytm BFS został zaimplementowany jako część Interface oraz wykorzystana została jedynie lista sąsiedztwa, prezentowana przez strukturę słownika języka Python.

Podczas oddawania projektu napotkano również problem z dwoma testami, jednakże możliwość poprawy algorytmu jest prosta.

*5. Efekt końcowy*

Projekt zawierał wykorzystanie konstrukcji list oraz dictionary comprehensions, wyrażenia lambda, kilka klas oraz klasę wyjątków. Całość została podzielona na moduły.

Algorytm przeszukiwania działał poprawnie, opierał się na znalezieniu najkrótszej trasy w oparciu o ilość stacji pośrednich.