

# **Projekt Sieć: Bartłomiej Jachowicz**

Plik **simulator.py** zawiera zdefiniowane przeze mnie dwie klasy reprezentujące metody routowania.

Pierwsza klasa **ShortPathRouter** wykorzystuje metodę wektora odległości. Klasa zawiera właściwość `distance_vec` przedstawiającą wektor odległości będący słownikiem, który dla każdego z wierzchołków w grafie (klucz) zawiera sąsiada przez którego prowadzi ścieżka do klucza oraz odległość do klucza. W klasie zdefiniowane są również metody `add_link` oraz `del_link` odpowiadające za dodawanie oraz usuwanie połączeń między routerami, odpowiednio aktualizując przy tym właściwość `distance_vec`. Tak jak było omawiane na zajęcia oraz zaproponowane w początkowej wersji pliku co wielokrotność liczby 5 zmiennej `self.tick` routery wysyłają do siebie informację z uaktualnioną zawartością właściwości `distance_vec`.

Druga klasa **GraphRouting** wykorzystuje metodę, w której dla każdego routera zapamiętany jest słownik `graph`, który dla danego klucza trzyma słownik z kluczem który jest drugim wierzchołkiem oraz wartością będącą parą (wiadomością) czy krawędź między dwoma wierzchołkami istnieje oraz z jakiego czasu wiadomość o tej krawędzi pochodzi. W tej klasie również odpowiednio zdefiniowane są metody `add_link` oraz `del_link` odpowiadające za dodawanie oraz usuwanie połączeń między routerami, odpowiednio aktualizując przy tym słownik `graph`. W tej klasie jeśli zaszła zmiana to co wielokrotność 20 zmiennej `self.tick` uaktualniany jest przez routery między sobą stan sieci. Wykorzystywana jest do tego funkcja `find_neighbour` wykorzystująca algorytm BFS w wersji iteracyjnej wykorzystujący strukturę danych kolejki.

Dodatkowo plik zawiera zdefiniowane przeze mnie testy, znajdujące się na końcu pliku. Testy zawierają grafy przedstawiające sieć routerów w których występują cykle oraz takie w których cykli nie. W testach wykorzystywane są metody `add_link` oraz `del_link` sprawdzające jak algorytmy radzą sobie z usuwaniem i dodawaniem krawędzi. Funkcję z testami jako argument przyjmują nazwę algorytmu który mają testować. By wykonać dany test, należy wywołać daną funkcję (`test1()`, `test2()`, `test3()` lub `test4()`) z odpowiednim argumentem.