

Algorytmy i struktury danych, Teleinformatyka, I rok

Raport z laboratorium nr: 8

Imię i nazwisko studenta: Mateusz Warzecha

nr indeksu: 410 846

1. W pole poniżej wklej najważniejszy (według Ciebie) fragment kodu źródłowego z zajęć (maksymalnie 15 linii).

```
1. def search_min_distances(elements:list, core:int):
2.     temp_id=0
3.     min_distance=[1000,0]
4.     for i in range(len(elements)):
5.         if elements[i][0]!=core:
6.             for j in range(len(elements)):
7.                 if elements[j][0]==core:
8.                     temp_id=j
9.                     distance=sqrt(((elements[temp_id][1]-elements[i][1]))**2 +
10. (elements[temp_id][2]-elements[i][2])**2)
11.                 if distance<min_distance[0]:
12.                     min_distance[0]=distance
13.                     min_distance[1]=elements[i][0]
13.     return min_distance
```

Wybrałem ten fragment kodu ponieważ zawiera się w nim kluczowa dla mojego algorytmu funkcja licząca odległości z konkretnego punktu do każdego innego i zwracająca liste zawierającą informacje który punkt znajduje się najbliżej oraz jego numer id

2. Podsumuj wyniki uzyskane podczas wykonywania ćwiczenia. Co ciekawego zauważyłeś? Czego się nauczyłeś? Jeśli instrukcja zawierała pytania, odpowiedz na nie. Do sprawozdania możesz dodać wykresy jeśli jest taka potrzeba.

Wykorzystując algorytm szukający najbliższego punktu dla podanego ,zaczynając od każdego kolejnego punktu okazuje się że najkrótsza trasa występuje dla punktu z indeksem 93 i jest to 894.8520254864372 jednostek. Jest to najkrotsza trasa możliwa do uzyskania używając algorytmu szukającego najbliższych sąsiadów. W porównaniu do zadania 1 gdzie idąc kolejno po indeksach miast ścieżka miała długość 5084.461369051878 różnica jest spora, ale to wciąż nie jest najbardziej optymalne rozwiązanie.