

Algorytmy i struktury danych, Teleinformatyka, I rok

Raport z laboratorium nr: 8

Imię i nazwisko studenta: Patryk Kurek

nr indeksu: 414887

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

```
def calculate_distance(city_coords, path):  
    distance = 0  
    num_cities = len(path)  
    for i in range(num_cities - 1):  
        city1 = path[i]  
        city2 = path[i+1]  
        x1, y1 = city_coords[city1]  
        x2, y2 = city_coords[city2]  
        distance += math.sqrt((x2 - x1)**2 + (y2 - y1)**2)  
    return distance
```

Uzasadnij swój wybór.

Fragment tego kodu zawiera definicję funkcji `calculate_distance`, która oblicza odległość między miastami na podstawie współrzędnych miast podanych jako pierwszy argument oraz ścieżki, która jest podana jako drugi argument. Jest to najważniejszy fragment kodu, ponieważ najmniejszy błąd we wzorze czy w implementacji może skutkować błędnym zwracaniem wartości dystansu dla poszczególnych miast.

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.

Algorytm, który wykorzystałem w moim kodzie to algorytm najbliższego sąsiada. Algorytm iteracyjnie dodaje do trasy najbliższe nieodwiedzone miasto na podstawie minimalnej odległości od bieżącego miasta. Przechodząc przez wszystkie miasta ostatecznie otrzymujemy trasę, która jest suboptymalna, ale daje wyniki bliskie rozwiązaniu optymalnego. Porównując ścieżkę z pliku oraz ścieżkę z algorytmu najbliższego sąsiada wyniki znacząco się różnią:
length of path from file: 5018.867569390631
length of path from nearest_neighbor: 992.3393569059313
Mamy więc tutaj poprawę dystansu o jakieś:
improvement: 505.7612130833176 %

Czasy:

time path from file: 8.368492126464844e-05

time nearest_neighbor algorithm: 0.0016405582427978516