

|  |  |
| --- | --- |
|  | **AKADEMIA GÓRNICZO-HUTNICZA** |

Raport i instrukcja z projektu

z przedmiotu

**Narzędzia komputerowe w rozwiązywaniu wybranych zagadnień matematyki wyższej i optymalizacji**

Elektronika i Telekomunikacja - Systemy Wbudowane, rok I studiów magisterskich

*Jakub Stelmach*

*Dominik Tendera*

*Jakub Ziobrowski*

14.06.2024r

Spis treści

[1. Temat projektu 3](#_Toc200835094)

[2. Wybrany algorytm 3](#_Toc200835095)

[3. Instrukcja 3](#_Toc200835096)

[4. Wnioski 4](#_Toc200835097)

# Temat projektu

Projekt nr 15

Niestety znów wykryto bazę Obcych na Ziemi! Jej kolejne komory tworzą graf z jednym tylko wejściem z zewnątrz. Niestety wszelkie metody zawiodły, pozostała już tylko jedna. Wybrany ochotnik (dlaczego to znów Ty?) musi wbiec do bazy i przebiec tyle komór ile zdoła w każdej kolejno mijanej porzucając bombę zegarową (w każdej może być tylko raz). Dostałaś mapę gniazda i znasz odległości czasowe pomiędzy komorami. Napisz algorytm, który ustali największą liczbę komór jakie zdołasz przebiegnąć w czasie T

# Wybrany algorytm

Zdecydowano się na wykorzystanie algorytmu DFS (przeszukiwanie w głąb) ze względu na zagłębianie się w graf, śledząc przy tym odwiedzone wierzchołki i poświęcony czas. W rozwiązaniu zaimplementowano również odcinanie gałęzi, gdy nawet w optymistycznym scenariuszu nie będzie można pobić dotychczasowego najlepszego wyniku.

# Instrukcja

Program realizujący podany algorytm został utworzony w języku Python, wersja jaką trzeba mieć zainstalowaną na komputerze to 3.8 i wyższe. W jednym folderze powinny się znaleźć plik „most\_nodes\_algorithm.exe”, który uruchomi nasz program oraz plik „graph\_values.txt” gdzie znajduje się nasz graf który sprawdza działanie algorytmu. Format pliku .txt to definicja limitu czasu, miejsce z którego zaczynamy trasę oraz tablica dwuwymiarowa (max. 20x20), gdzie kolumny i wiersze to wierzchołki, a wartości to czas przejścia między poszczególnymi punktami. Poniżej przykładowa struktura pliku:

Defined Time: 100

Which node is the start: 1

1 2 3 4 5 6 7 8 9 10 11

1 0 34 78 0 65 12 0 45 0 19 83

2 56 0 0 23 0 88 14 0 90 0 47

3 0 42 0 51 77 0 0 0 29 65 0

4 73 0 0 0 38 0 91 27 0 33 55

5 0 19 0 0 0 60 79 0 0 0 0

6 21 0 0 0 18 0 53 92 0 0 41

7 0 38 0 0 0 0 0 0 0 96 0

8 44 0 85 0 0 0 0 0 22 0 70

9 0 0 0 0 0 33 0 0 0 0 0

10 0 71 60 0 0 0 0 93 48 0 17

11 50 0 0 84 0 0 65 0 0 0 0

Dla zobrazowania, czas który mamy do dyspozycji to 100 jednostek, zaczynamy od punktu [1, 1] i dostanie się do punktu numer 3 z punktu o numerze 2 zajmie nam 42 jednostki czasu. Jako rezultat naszego programu otrzymamy w oknie konsoli wybraną trasę zawierającą najwięcej odwiedzonych wierzchołków oraz wykorzystany czas:

Obraz zawierający tekst, Czcionka, zrzut ekranu, czarne

Zawartość wygenerowana przez AI może być niepoprawna.

*Rys 1. Rezultat wykonania programu*

# Wnioski

Realizacja projektu pozwoliła pogłębić umiejętności programowania w języku Python oraz zrozumienie teorii tworzenia algorytmów grafowych.