

IMIĘ I NAZWISKO Mariusz Dajczak
NR INDEKSU 200403
TERMIN czwartek 10:00-12:35
DATA 15.05.2014

PROJEKTOWANIE ALGORYTMOW I METODY SZTUCZNEJ INTELIGENCJI

SPRAWOZDANIE Z LABORATORIUM

Metody przeszukiwania grafu

1. Wstęp

Graf to podstawowa struktura danych, posiada on wierzchołki i krawędzie. Każda krawędź łączy 2 wierzchołki. W mojej implementacji graf jest reprezentowany w pamięci komputera jako macierz sąsiedztwa, gdzie $x=0$ to brak połączenia, natomiast $x>0$ to połączenie oraz waga krawędzi.

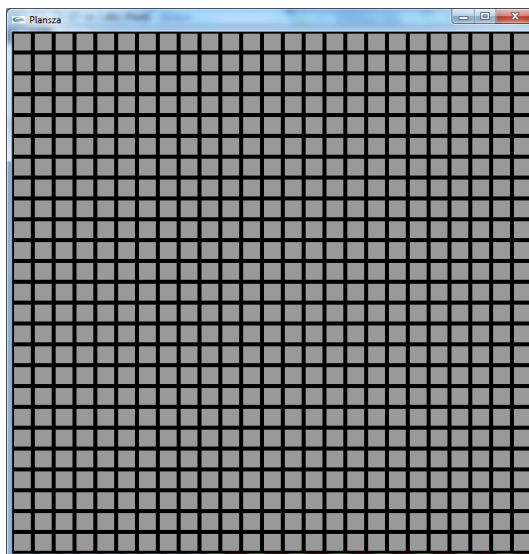
Istnieje wiele metod służących do znajdowania drogi w grafie. Naszym zadaniem było zaimplementowanie 3 algorytmów:

- przeszukiwanie w głąb - DFS
- przeszukiwanie w szerz - BFS
- algorytm A*

W sprawozdaniu przedstawię różnicę w szybkości działania tych algorytmów.

2. Program do wizualizacji

W celu wizualizacji poszukiwań drogi z punktu A do punktu B stworzyłem prosty program graficzny, który wyświetla znaną drogę oraz które wierzchołki były odwiedzone podczas szukania.



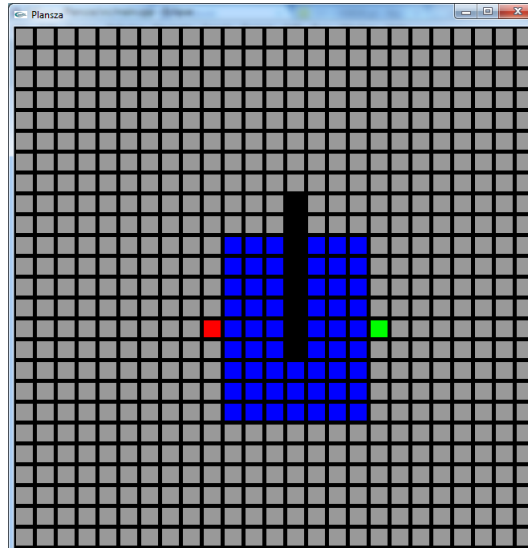
Rysunek 1. Graficzna reprezentacja grafu

Mój graf składa się z 625 wierzchołków (plansza 25x25 pól). Każde pole(wierzchołek) połączone jest ze swoim górnym, dolnym, lewym oraz prawym sąsiadem. Oczywiście wyjątkiem są krawędzie planszy. Każde takie połączenie ma wagę 1. Dodając pole brązowe zmieniamy wszystkie wagi krawędzi wychodzących z danego wierzchołka na wartość 2, niebieskie pole zmienia je na wartość 10. Zielone pole oznacza punkt startu, czerwone punkt docelowy. Pole żółte sygnalizuje, że wierzchołek był przez algorytm odwiedzony. Dodatkowo pole czarne symbolizuje przeszkodę nie do przejścia, wierzchołek jest wtedy zablokowany.

3. Wyniki pomiarów

3.1. Plansza testowa

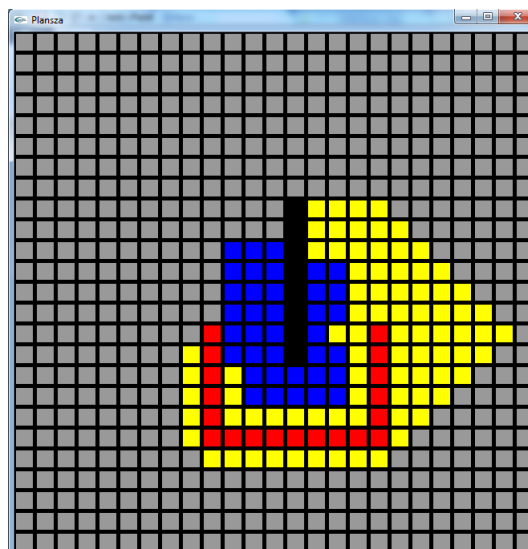
Stworzyłem testową planszę, na której będę sprawdzał szybkość i skuteczność działania algorytmów przeszukiwania.



Rysunek 2. Plansza testowa

3.2. Algorytm A*

Algorytm A* poradził sobie z zadaniem, wyznaczył najkrótszą drogę omijając niebieski obszar. Funkcją heurystyczną jest tutaj metoda typu Manhattan.



Rysunek 3. Przeszukiwanie algorytmem A*

Ilość odwiedzonych pól : 135

Koszt : 18

3.3. Algorytm BFS

Algorytm BFS znalazł drogę zawierającą minimalną ilość wierzchołków, jednak nie jest ona optymalna.



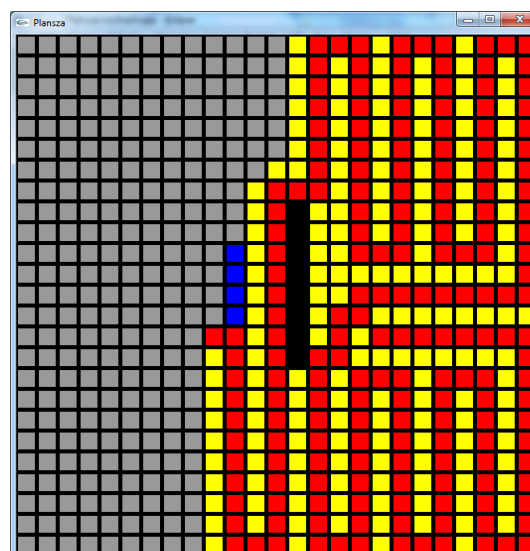
Rysunek 4. Przeszukiwanie algorytmem BFS

Ilość odwiedzonych pól : 12069

Koszt : 93

3.4. Algorytm DFS

Algorytm DFS odnalazł bardzo długą i krętą drogę, jednak wykonał to bardzo szybko.



Rysunek 5. Przeszukiwanie algorytmem DFS

Ilość odwiedzonych pól : 351

Koszt : 470

3.5. Zestawienie

Algorytm	Odwiedzone wierzchołki	Koszt
A*	135	18
BFS	12069	93
DFS	351	470

Tabela 1. Ilość odwiedzonych wierzchołków i koszt dojścia dla różnych algorytmów

4. Wnioski

Po przeprowadzeniu symulacji widzimy znaczną przewagę algorytmu A*, który nie tylko znalazł najkrótszą drogę, ale również zrobił to najszybciej. Algorytm BFS działał bardzo długo, jednak pomimo to droga, którą znalazł nie jest optymalna. DFS poradził sobie bardzo szybko z zadaniem jednak droga, jaką wyznaczył jest nie do przyjęcia. Znaczną przewagę algorytm A* zawdzięcza heurystyce, dzięki której zawsze prusza się w kierunku celu. Pozostałe algorytmy poszukują wierzchołka docelowego po omacku, bo nie mają heurystyki.