

IMIĘ I NAZWISKO Tomasz Piotrowski
NR INDEKSU 200524
DATA 24.04.2014

PROJEKTOWANIE ALGORYTMÓW I METODY SZTUCZNEJ INTELIGENCJI

SPRAWOZDANIE Z LABORATORIUM **GRAF-przyszukiwanie wszerz i w głąb**

1. Wstęp

W informatyce grafem nazywamy strukturę $G=(V, E)$ składającą się z węzłów (wierzchołków, oznaczanych przez V) wzajemnie połączonych za pomocą krawędzi (oznaczonych przez E). Grafy dzielimy na grafy skierowane i nieskierowane:

Celem tego ćwiczenia jest zaimplementowanie struktury grafu oraz przetestowanie działania algorytmów przechodzenia w szerz .

2. Graf

W literaturze można znaleźć dwie propozycje implementacji grafu. Implementacja z pomocą list sąsiedztwa lub za pomocą macierzy sąsiedztwa. Graf zaimplementowany przezemnie opiera się na dynamicznej tablicy oraz vectorze. Wierzchołki grafu przechowywane są w tablicy dynamicznej a krawędzie w listach sąsiedztwa. Implementacja przezemnie wybrana charakteryzuje się korzystniejszą złożonością obliczeniową $O(V + E)$. Macierz sąsiedztwa posiada złożoność obliczeniową $O(V^2)$. Program przezemnie napisany pozwala na tworzenie grafu za pomocą dostarczonego MENU, lub odpowiednio wyedytowanych plików *.txt.

Do generowania pliku tekstowego została napisana odpowiednia funkcja. Wygenerowany graf charakteryzuje się krótkimi połączeniami, każdy wierzchołek łączy się z wierzchołkiem zapisanym maksymalnie o 6 pozycji dalej w tablicy. Dzięki takiej budowie grafu możemy przeprowadzić testy wyszukiwania w których funkcja wraz z większą odległością między wierzchołkami gwarantuje większą ilość połączeń do sprawdzenia.

3. Przeszukiwanie wszerz

Przeszukiwanie wszerz (ang. Breadth-first search, w skrócie BFS) – jeden z algorytmów przeszukiwania grafu. Przechodzenie grafu rozpoczyna się od zadanego wierzchołka s i polega na odwiedzeniu wszystkich osiągalnych z niego wierzchołków. Wynikiem działania algorytmu jest także drzewo przeszukiwania wszerz o korzeniu w s , zawierające wszystkie wierzchołki do których prowadzi droga z s .

Algorytm działa prawidłowo zarówno dla grafów skierowanych jak i nieskierowanych. Algorytm przechodzenia wszerz jest kompletny. Jeśli istnieje droga między zadanymi wierzchołkami, zostanie ona zawsze odnaleziona.

Złożoność czasowa algorytmu związana jest z ilością krawędzi oraz wierzchołków jakie musi przejść algorytm. W najgorszym przypadku jest to $O(|V| + |E|)$

Zastosowanie algorytmu:

- odnalezienie wszystkich połączonych węzłów w grafie.
- odnalezienie najkrótszej drogi między dwoma wierzchołkami.
- sprawdzenie czy graf jest dwudzielny

4. wyszukiwanie drogi za pomoca bfs

W programie została zaimplementowana funkcja pozwalająca na odnalezienie drogi za pomoca przeszukiwania w szerz. Droga znaleziona ta metoda jest najkrótsza z możliwych dróg. Algorytm nie uwzględnia jednak wag połączeń co powoduje jego niedoskonałość.

5. Przeszukiwanie w głąb

Przeszukiwanie w głąb (ang. Depth-first search, w skrócie DFS) – jeden z algorytmów przeszukiwania grafu. Przeszukiwanie w głąb polega na badaniu wszystkich krawędzi wychodzących z podanego wierzchołka. Po zbadaniu wszystkich krawędzi wychodzących z danego wierzchołka algorytm powraca do wierzchołka, z którego dany wierzchołek został odwiedzony[1].

Złożoność czasowa algorytmu również wynosi $O(|V| + |E|)$ ponieważ tak jak i w przechodzeniu wszerz algorytm odwiedza wszystkie białe wierzchołki

Złożoność pamięciowa przeszukiwania w głąb w przypadku drzewa jest o wiele mniejsza niż przeszukiwania wszerz, gdyż algorytm w każdym momencie wymaga zapamiętania tylko ścieżki od korzenia do bieżącego węzła, podczas gdy przeszukiwanie wszerz wymaga zapamiętywania wszystkich węzłów w danej odległości od korzenia, co zwykle rośnie wykładniczo w funkcji długości ścieżki.

Algorytm przechodzenia w głąb jest wykorzystywany jako podprogram w innych algorytmach działających na grafach.

- do wyznaczania silnych spójnych składowych grafu skierowanego
- w algorytmie sortowania topologicznego skierowanego grafu acyklicznego

6. wyszukiwanie drogi za pomoca dfs

W programie została zaimplementowana również funkcja wyszukiwania drogi za pomoca algorytmu dfs. Funkcja ta znajduje drogę, jednak w przeciwieństwie do wyszukiwania za pomoca algorytmu bfs nie jest to droga najkrótsza. Funkcja również nie uwzględnia wag połączeń.

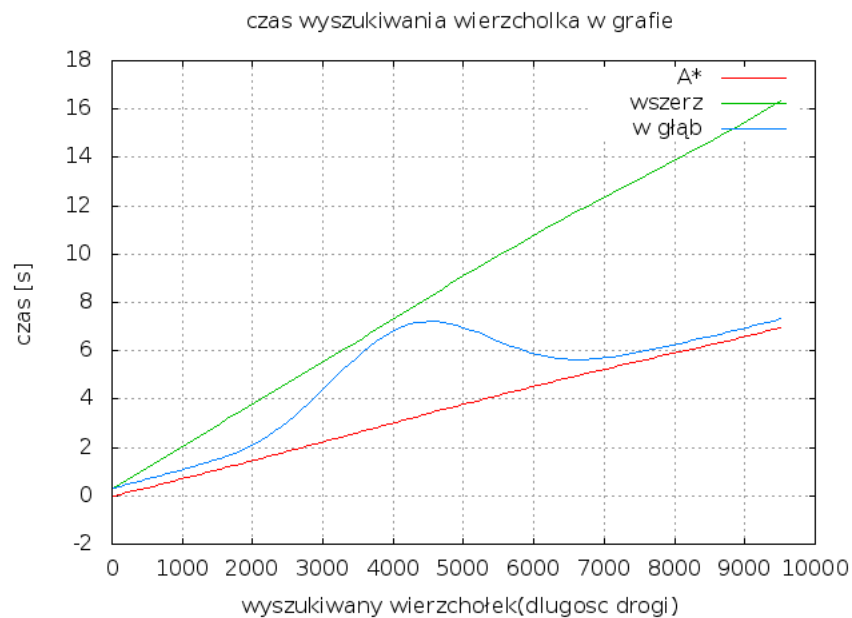
7. wyszukiwanie A*

W programie implementacja A* została oparta na vectorze z biblioteki STL, oraz tablicy dynamicznej która jest zwracana przez funkcję wyszukiwania.

W implementacji zapisana została heurystyka euklidesowa oraz manhatan. Za pomocą drobnej zmiany w pliku graf.cpp można wybrać sposób wybierania heurystyki. W obliczeniu heurystyki założono że graf jest rozpięty na planszy o wymiarach 100x100 pól. Dzięki temu założeniu możemy wyliczyć pozycję x i y i następnie wartość funkcji heurystycznej.

8. Test działania algorytmów wyszukiwania

Testy przeprowadzono na grafie losowo wygenerowanym, Za pomocą funkcji opisanej w punkcie 1. Droga w grafie zostaje wyszukiwana 10000rotnie by można było zmierzyć czas działania tej operacji.



odległość	A*	bfs(wszerz)	dfs(w głąb)
10	-2.81318e-42	0.34	0.33
510	0.33	1.2	0.7
1010	0.7	2.07	1.09
1510	1.09	2.93	1.46
2010	1.46	3.81	1.83
2510	1.83	4.67	2.22
3010	2.22	5.56	2.6
3510	2.6	6.42	3.01
4010	3.01	7.3	13.831
4510	3.53	8.17	13.31
5010	3.81	9.3	4.18
5510	4.18	9.97	4.52
6010	4.52	10.82	4.9
6510	4.9	11.73	5.29
7010	5.29	12.58	5.68
7510	5.68	12.97	5.85
8010	5.85	13.82	6.24
8510	6.24	14.64	6.58
9010	6.58	15.48	6.97
9510	6.97	16.34	7.32

Tabela 1. Czas wyszukiwania

9. Wnioski

Z analizy tabeli wynika jasno że najlepszy czas wyszukiwania osiąga algorytm A*. Jest on najszybszy niezależnie od długości ścieżki i ilości wierzchołków na niej.

W przypadku wyszukiwania w głąb czas wyszukiwania jest różny. Zależy on od tego w której kolejności przeszukana zostanie ścieżka na której znajduje się wyszukiwany wierzchołek.

Algorytm przeszukiwania w szerz jest najwolniejszy. Czas jego działania narasta wraz ze wzrostem problemu, rozrostem grafu.

Zbadana została również długość ścieżek wyznaczonych przez algorytmy. Okazuje się że długość ścieżki wyznaczonej przez Algorytm A* oraz algorytm wyszukiwania w szerz jest taka sama, Jednak różnią się wierzchołkami ponieważ algorytm A* uwzględnia wartości wag. Ścieżka wyznaczona przez algorytm wyszukiwania w głąb jest dużo dłuższa.

10. Bibliografia

— Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein:
Wprowadzenie do algorytmów.