

IMIĘ I NAZWISKO Tomasz Piotrowski
NR INDEKSU 200524
TERMIN czwartek 10:00-12:35
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 MEnuU, lub odpowiednio wyedytowanych plików *.txt.

3. Przeszukiwanie wszere

Przeszukiwanie wszere (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 wszere o korzeniu w s , zawierające wszystkie wierzchołki do których prowadzi droga z s .

Algorytm działa prawidłowo zarówno dla grafów skierowanych jak i nieskierowanych. Algorytm przechodzenie wszere 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 wagi 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 przechodzeni w głąb jest wykorzystywany jako pod program w innych algorytmach działających na grafach.

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

6. wyszukiwanie drogi za pomocą dfs

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

7. Bibliografia

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