

GRAFY - Laboratorium nr 8 z PAMSI

Justyna Klijewska

11 05 2014

1 WSTĘP

Celem ćwiczenia było zaimplementowanie grafu nieskorelowanego:

DodajWierzcholek (v)

DodajKrawedz ($v1, v2, C$)

UsunWierzch

UsunKrawedz

CzyPolaczone($v1, v2$)

Sasiedztwo (v)

Dodatkowo należało zaimplementować Breadth first search oraz Depth first search.

2 TEORIA

Graf (ang. graph) jest strukturą danych do reprezentacji obiektów, pomiędzy którymi występują różne zależności. Grafem nieskierowanym nazywamy graf w którym kierunek przebywania krawędzi od jednego wierzchołka do drugiego nie ma znaczenia. Istnieją dwa standardowe reprezentacje grafu - za pomocą list sąsiedztwa oraz macierzy sąsiedztwa. Istnieją dwa sposoby przeszukiwania takich grafów:

- Przeszukiwanie wszerek (BFS)

Jako dane mamy graf i wyróżniony wierzchołek początkowy s . W metodzie BFS bada się systematycznie krawędzie w celu odwiedzenia każdego wierzchołka, który jest osiągalny z naszym wierzchołkiem s . W tym samym momencie obliczane są odległości. W wyniku otrzymujemy drzewo przeszukiwania wszerek o korzeniu w s , które zawiera wszystkie osiągalne wierzchołki. Algorytm ten działa dla grafów skierowanych i nieskierowanych. Dodatkowo granica między wierzchołkami odwiedzonymi i nieodwiedzonymi jest przekraczana jednocześnie na całej szerokości.

Algorytm BFS:

1. Utwórz kolejkę.
2. Zapisz do kolejki wierzchołek początkowy.
3. Oznacz wierzchołek s jako odwiedzony.
4. Dopóki kolejka jest niepusta wykonuj:
 - 4.1. Pobierz z kolejki wierzchołek i nazwij go s .
 - 4.2. Jeśli S jest poszukiwanym wierzchołkiem końcowym, to zwróć SUK-

CES i zakończ algorytm.

4.3. Jeśli s nie jest poszukiwanym wierzchołkiem końcowym, to zapisz do kolejki wszystkie nieodwiedzone wierzchołki sąsiadujące z s .

5. Zwróć BRAK ROZWIĄZANIA i zakończ algorytm.

- Przeszukiwanie w głąb (DFS)

Przy tym przeszukiwaniu badane są krawędzie ostatnio odwiedzonego wierzchołka v , z którego jeszcze wychodzą zbadane krawędzie. Po zbadaniu wszystkich krawędzi przeszukiwanie "wraca" do wierzchołka, z którego v został odwiedzony. Proces ten jest powtarzany do momentu dopóki wszystkie wierzchołki nie zostaną odwiedzone. W przeciwieństwie do BFS podgraf poprzedników może zawierać wiele drzew.

Algorytm DFS:

1. Utwórz stos.

2. Zapisz do stosu wierzchołek początkowy.

3. Oznacz wierzchołek v jako odwiedzony.

4. Dopóki stos jest niepusty wykonuj:

4.1. Pobierz ze stosu wierzchołek i nazwij go v . 4.2. Jeśli v jest poszukiwanym wierzchołkiem końcowym, to zwróć SUKCES i zakończ algorytm.

4.3. Jeśli v nie jest poszukiwanym wierzchołkiem końcowym, to dodaj do stosu wszystkie nieodwiedzone wierzchołki sąsiadujące z S .

5. Zwróć BRAK ROZWIĄZANIA i zakończ algorytm.

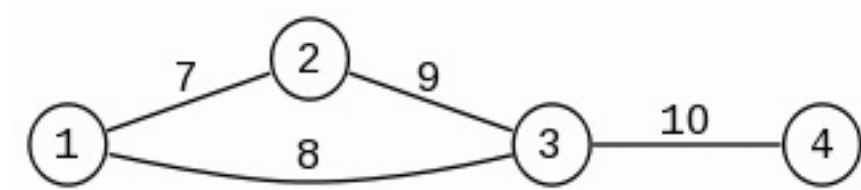
3 IMPLEMENTACJA

Graf zaimplementowano za pomocą macierzy sąsiedztwa (alternatywą może być lista sąsiedztwa). W tym przypadku zakładamy że wierzchołki są ponumerowane w dowolny sposób. Ta reprezentacja używana jest do grafów z wagami. Dodatkowo wymaga ona $O(n^2)$ pamięci (niezależnie od liczby krawędzi w tym grafie).

Algorytmy DFS i BFS zostały opisane powyżej i tak właśnie zostały zaimplementowane.

4 PRZYKŁADY

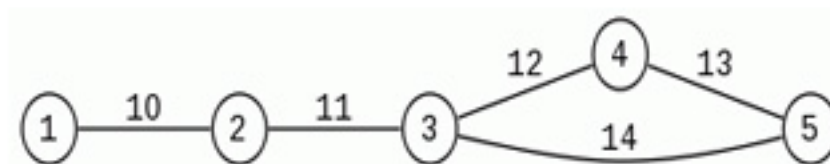
Przykład 1



W powyższym grafie mamy poniższe wyszukiwania:

- Przeszukiwanie wszerz: 1 2 3 4
- Przeszukiwanie w głąb: 1 3 4 2

Przykład 2



W powyższym grafie mamy poniższe wyszukiwania:

- Przeszukiwanie wszerz: 1 2 3 4 5
- Przeszukiwanie w głąb: 1 2 3 5 4

Grafy zostały wygenerowane na stronie <http://www.algorytm.org/narzedzia/edytor-grafow.html>.

5 WNIOSKI

Można znaleźć wiele różnych zastosowań dla grafów. Szczególny ich rodzaj zwany drzewami można zastosować do reprezentacji hierarchii np.: w mistrzostwach sportowych.

Dodatkowo mają swoje zastosowania również w środowisku komputerowym. Jako graf można przedstawić sieć dróg (skrzyżowania to wierzchołki, a ulice są krawędziami). Taka reprezentacja sprzyja znalezieniu najlepszej (najkrótszej drogi) z jednego punktu do drugiego). Z tego typu rozwiązania korzysta np.: GPS, czy serwis jakdojade.pl.

Grafy są też stosowane w robotyce - przedstawienie sieci komputerowych w postaci grafów pozwoliło na stworzenie oprogramowania, które powodowało, że maszyny mogły bez udziału człowieka odnaleźć trasę w trudnym terenie. To tylko nieliczne przykłady zastosowań algorytmów grafów.