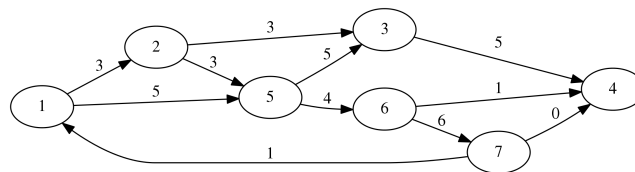


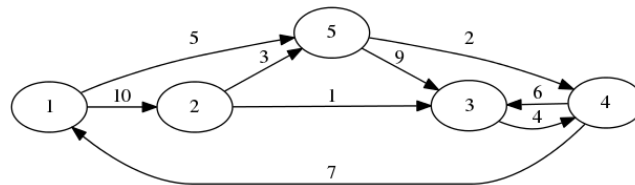
Grafy - algorytmy wyszukiwania

Marcin Ochman

11 maja 2014



Rysunek 1: Graf, który został użyty do testowania algorytmów przeszukiwania



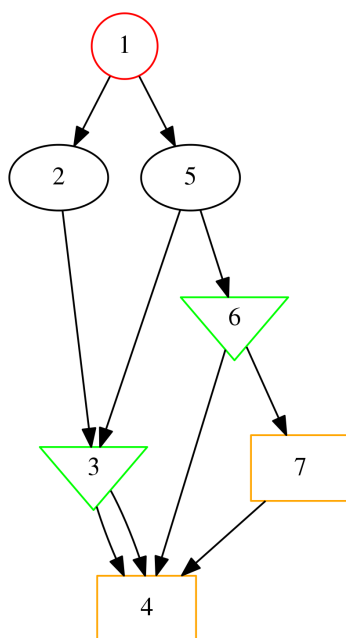
Rysunek 2: Drugi graf, który został użyty do testowania algorytmów przeszukiwania

1 Opis problemu

Jednym z najczęstszych problemów, z którymi można się spotkać w świecie grafów to problem wyszukiwania ścieżki z wierzchołka startowego do wierzchołka końcowego. Dodatkowo, najczęściej interesuje nas najkrótsza ścieżka, ale czasami wystarczy, że algorytm znajdzie tylko taką ścieżkę, która zaprowadzi nas z określonego miejsca do punktu docelowego. Na laboratorium zaimplementowałem cztery różne algorytmy przeszukiwania, przy czym ostatni znajduje optymalną ścieżkę tzn. koszt jest najmniejszy. Są to:

- DFS - deep first search
- BFS - breadth first search
- Best First Search
- A*

2 Opis poszczególnych algorytmów



Rysunek 3: Drzewo, które zostało zbudowane w oparciu o graf z Rysunku 3. Każdy poziom został wyróżniony różnym kształtem oraz kolorem

Każdy z algorytmów działa nieco inaczej. Postaram się w kilku słowach opisać, jak funkcjonuje każdy z nich.

2.1 DFS - Deep First Search

Jak sama nazwa wskazuje algorytm przeszukuje graf w głąb tzn. najpierw brane są pod uwagę wierzchołki, które są sąsiadami danego wierzchołka - procedura wykonuje się rekurencyjnie. Na drzewie na Rysunku 3 przeszukiwać można powiedzieć, że algorytm przechodzi najpierw w dół, następnie wraca do wierzchołka i powtarza procedurę. Każdy poziom jest odwiedzany kilkakrotnie.

2.2 BFS -Breadth First Search

Polska nazwa tego algorytmu to wyszukiwanie wszerek. Polega on na odwiedzaniu wierzchołków należących do tego samego poziomu, a następnie schodzi on poziom niżej.

2.3 Best First Search

Jest to algorytm zachłanny - odwiedza pierwszy ten wierzchołek, który spełnia pewne kryterium np. minimalne waga połączenia z poprzednim wierzchołkiem.

2.4 A*

Jest to algorytm oparty na funkcji heurystycznej. Na jej podstawie potrafi dobrać, która ścieżka jest najlepsza w odpowiednim momencie. Dla określonych funkcji heurystycznych mamy gwarancję, że zostanie znaleziona optymalna ścieżka. Algorytm ten jest często używany w grach. Dla funkcji heurystycznej, która zawsze zwraca 0, algorytm ten sprowadza się do algorytmu Dijkstry.

3 Porównanie poszczególnych algorytmów w praktyce

Do porównania działania algorytmów posłużyłem się grafem z Rysunku 2. Zadaniem było znalezienie ścieżki z wierzchołka 1 do 4. Poszczególne ścieżki, dla różnych algorytmów zostały przedstawione na Listingu

```
1 Graf nr 1:
2
3 A*:
4 1->5->6->4
5 DFS:
6 1->5->6->7->4
7 BFS:
8 1->2->3->4
9 Best First Search:
10 1->2->3->4
11
12 Graf 2:
13
14 A*:
15 1->5->4->3
16 DFS:
17 1->5->4->3
18 BFS:
19 1->2->3
20 Best First Search:
21 1->5->4->3
```

Listing 1: Wyniki poszukiwania ścieżek dla poszczególnych algorytmów

4 Wnioski oraz uwagi

Na podstawie przeprowadzonych testów można wysnuć następujące wnioski:

- Algorytm A* faktycznie znalazł najkrótszą ścieżkę, funkcja heurystyczna zawsze zwracała 0
- Algorytm Best First Search za każdym razem wybierał drogę, która była najmniej kosztowna

- Budowa grafu ma duży wpływ na to, który algorytm DFS, BFS oraz Best First Search będzie lepszy tzn. pokaże lepszą ścieżkę