

Algorytm A^* - Laboratorium nr 9 z PAMSI

Justyna Klijewska

21 05 2014

1 WSTĘP

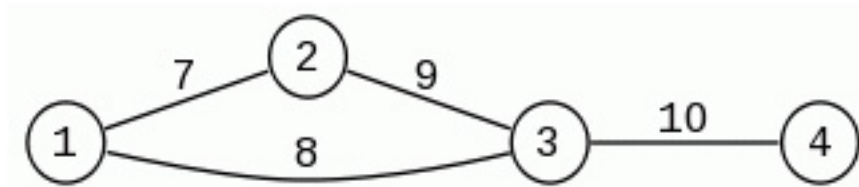
Celem ćwiczenia było zaimplementowanie w istniejącym programie algorytmu wyszukiwania A^* i zmierzenie drogi oraz czasów poszczególnych wyszukiwań. Program był pisany w środowisku Windows i została zrealizowana wersja na ocenę 5.

2 TEORIA

Algorytm A^* – algorytm heurystyczny znajdowania najkrótszej ścieżki w grafie ważonym z dowolnego wierzchołka do wierzchołka spełniającego określony warunek zwany testem celu. Algorytm jest zupełny i optymalny, w tym sensie, że znajduje ścieżkę, jeśli tylko taka istnieje, i przy tym jest to ścieżka najkrótsza. Stosowany głównie w dziedzinie sztucznej inteligencji do rozwiązywania problemów i w grach komputerowych do imitowania inteligentnego zachowania.

3 PRZYKŁADY

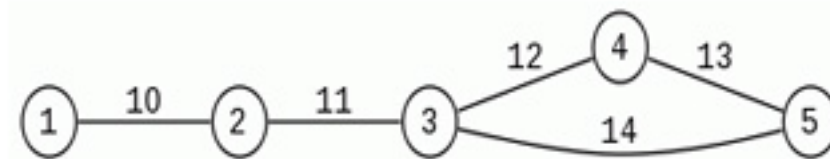
Przykład 1



W powyższym grafie mamy poniższe wyszukiwania:

- Przeszukiwanie wszerz: 1 2 3 4 czas: 3056 ns
- Przeszukiwanie w głąb: 1 3 4 2 czas: 3779 ns
- Przeszukiwanie A^* : 1 3 4 czas: 16694 ns

Przykład 2

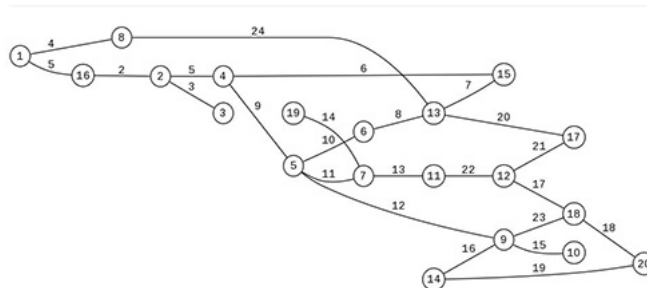


W powyższym grafie mamy poniższe wyszukiwania:

- Przeszukiwanie wszerz: 1 2 3 4 5 czas: 4578 ns
- Przeszukiwanie w głąb: 1 2 3 5 4 czas: 3947 ns
- Przeszukiwanie A^* : 1 2 3 5 czas: 97 ns

Przykład 3

W powyższym grafie mamy poniższe wyszukiwania:



- Przeszukiwanie wszerek: 1 8 16 13 2 6 15 17 3 4 5 12 7 9 11 18 19 10 14 20 czas: 22238 ns
- Przeszukiwanie w głąb: 1 16 2 4 15 13 17 12 18 20 14 9 10 5 7 19 11 6 8 3 czas: 22231 ns
- Przeszukiwanie A^* : 1 16 2 4 5 9 14 20 czas: 348 ns

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

4 WNIOSKI

Można przyjąć, że A^* jest najbardziej efektywnym (najszybszym) algorytmem wyszukiwania ścieżki w grafie. Dzieje się tak dlatego, że zamiast przeszukiwać ślepo po kolei cały graf możemy najmniej prawdopodobną ścieżkę odrzucić (a co się z tym wiąże - znaleźć szybko najkrótszą ścieżkę). Oczywiście od każdej reguły są wyjątki. Jeżeli szukamy najlepszego pod względem wydajności to przy małych ilościach danych BFS i DFS są efektywniejsze. Podobna sytuacja jest wtedy, kiedy szukamy algorytmu najbardziej niezawodnego. W tym przypadku lepiej sprawdzają się algorytmy, które zawsze znajdą ścieżkę, jeśli taka istnieje (np.: DFS).

Jak widać w powyższych przykładach kiedy nasz graf ma wiele wierzchołków i krawędzi to algorytm A^* działa o wiele szybciej niż pozostałe algorytmy.