

PAMSI – testowanie algorytmów wyszukiwania w grafie

Piotr Wilkosz

06/05/2014

1 Wstęp

Celem ćwiczenia było przetestowanie implementacji algorytmów wyszukiwania ścieżki w grafie. Graf został stworzony na podstawie listy incydencji. Przetestowano następujące algorytmy:

- Przeszukiwanie włąb - DFS
- Przeszukiwanie wszere - BFS
- Przeszukiwanie grafu poczynając od najkrótszej ścieżki - Best - First Search
- Algorytm A*

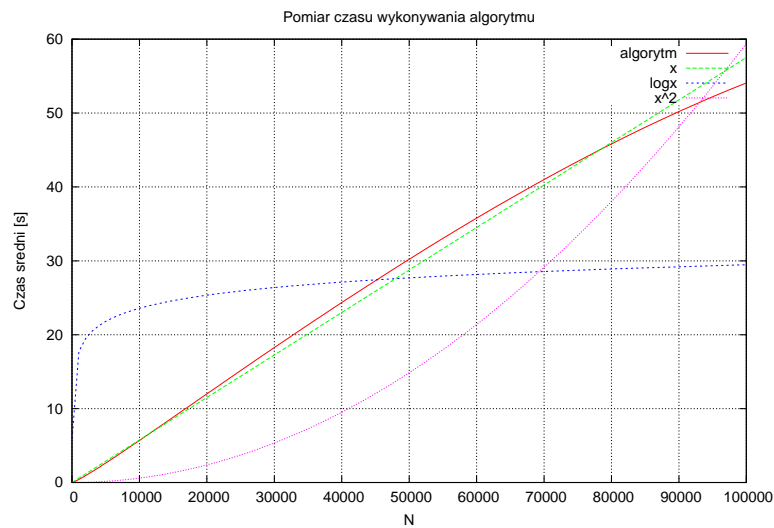
Zadaniem było zmierzenie czasu wykonywania operacji wyszukania losowego elementu w grafie.

2 Wyniki pomiarów

1. DFS

Tablica 1: Pomiar czasu przeszukiwania włąb w grafie

N	czas	odchylenie
10	0.00110325	0.000130398
100	0.0274422	0.00285908
1000	0.134149	0.0233129
10000	3.41307	1.98324
50000	36.1375	11.0835
100000	54.0686	16.6036



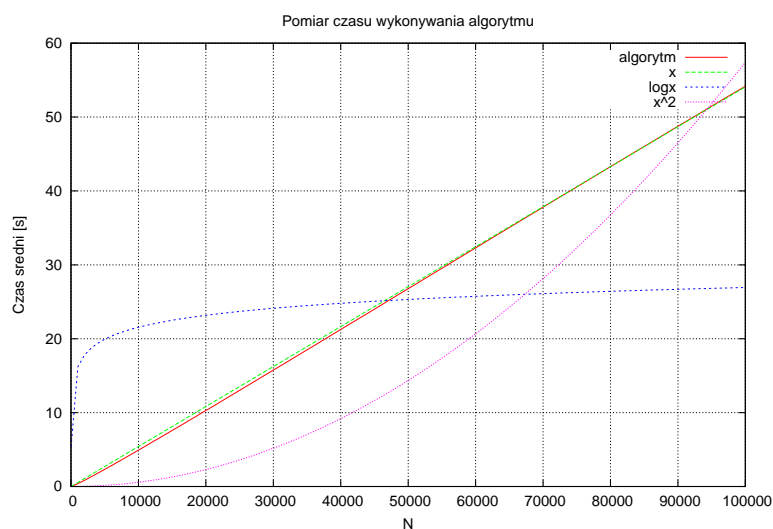
Rysunek 1: Wykres do tabeli nr 1

Na podstawie wykresu1 i tabeli 1 czas operacji przeszukiwania wgląb szacuje się jako liniowy. Duże odchylenie od średniego czasu przeszukiwania świadczy o tym, iż zarówno przypadek pesymistyczny, jak i optymistyczny przypada z jendakowym prawdopodobieństwem, co powoduje znaczące zróżnicowanie.

2. BFS

Tablica 2: Pomiar czasu przeszukiwania wszecz w grafie

N	czas	odchylenie
10	0.000492332	4.54287e-05
100	0.00385789	0.000494668
1000	0.116186	0.0625103
10000	4.10229	1.92677
50000	27.1518	10.3526
100000	54.1763	20.0539



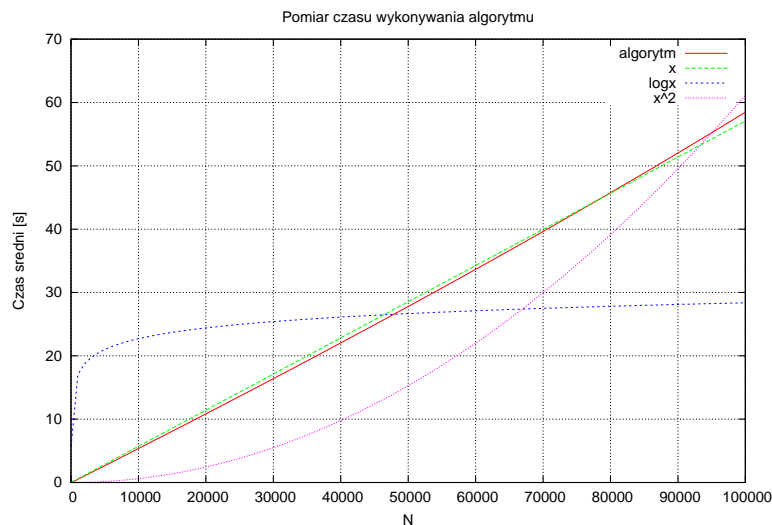
Rysunek 2: Wykres do tabeli nr 2

Na podstawie wykresu 2 i tabeli 2 czas operacji przeszukiwania wszerek szacuje się jako liniowy. Podobnie jak w przypadku wcześniejszym, zaobserwować można spore odchylenie, jednak uśrednienie wyników daje czas proporcjonalny do sumy liczb wierzchołków i krawędzi w grafie.

3. Best - First Search

Tablica 3: Pomiar czasu przeszukiwania typu best - first w grafie

N	czas	odchylenie
10	0.000659847	6.09969e-05
100	0.0108403	0.000987436
1000	0.358854	0.217304
10000	5.71089	2.71159
50000	25.7899	13.9438
100000	58.4636	24.0813



Rysunek 3: Wykres do tabeli nr 3

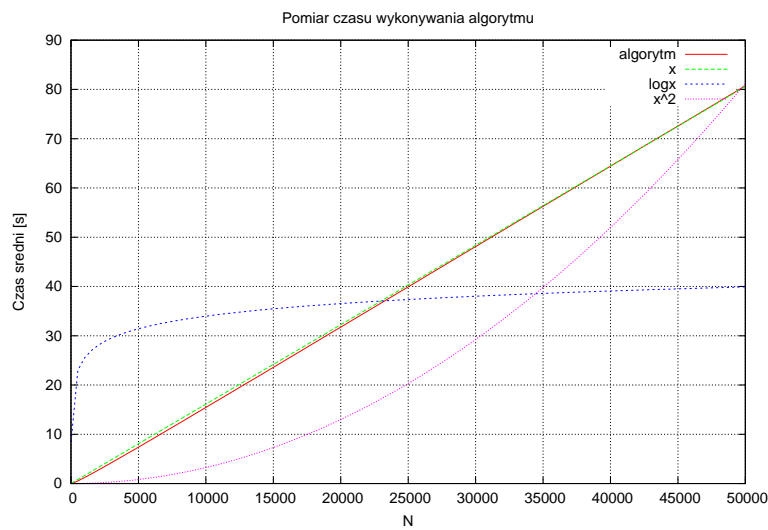
Z danych z wykresu 3 i tabeli 3 wyciągnąć podobne wnioski jak w przypadku dwóch poprzednich algorytmów. Czas wykonania operacji jest liniowy.

4. Algorytm A*

Algorytm A* służy do znajdowania najkrótszej ścieżki w grafie. W przeciwieństwie do powyższych algorytmów, nie podaje dowolnej ścieżki, lecz możliwie najkrótszą. Oczywiście dokładność tego algorytmu zależy od przyjętej heurystyki, im szacowana odległość jest bliższa rzeczywistej, tym lepszą ścieżkę wskaże algorytm. Na potrzeby ćwiczenia zaimplementowano reprezentację grafu, która wspomagana była przez rozmieszczenie wierzchołków na płaszczyźnie kartezjańskiej. Do wyznaczania heurystyki użyto tzw. metryki typu Manhattan. W metryce tej odległość wyznacza się poprzez sumę odległości przebytych w linii prostej i poziomej. Ponadto długości ścieżek wybierane są losowo, gdzie najkrótsza możliwa odległość jest równa szacowanej odległości pomiędzy dwoma punktami, co zapobiega przeszacowaniu heurystyki.

Tablica 4: Pomiar czasu działania algorytmu A* w grafie

N	czas	odchylenie
10	0.00104789	0.000782535
100	0.021713	0.00951804
1000	0.177888	0.13306
10000	15.6215	31.3469
50000	80.7123	88.6142



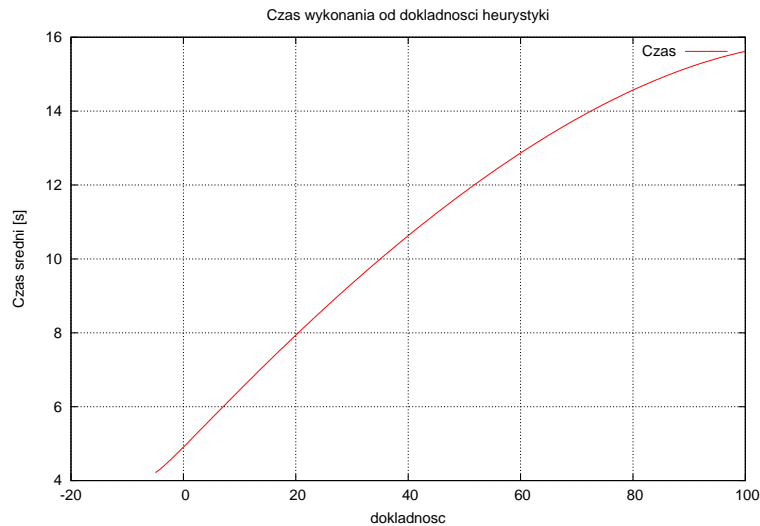
Rysunek 4: Wykres do tabeli nr 4

Z danych z wykresu 4 i tabeli 4 można wyciągnąć podobne wnioski jak w przypadku dwóch poprzednich algorytmów. Czas wykonania operacji jest wprost proporcjonalny do sumy ilości węzłów oraz krawędzi.

2.1 Zależność szybkości algorytmu A* od dokładności heurystyki

Działanie algorytmu A* w istotny sposób zależy od jakości oszacowania drogi potrzebnej do przebycia pomiędzy danymi punktami. Poniżej umieszczono wykres przedstawiający szybkość działania algorytmu A* w zależności od stopnia dokładności heurystyki. Jak już wcześniej wspomniano, długość ścieżki pomiędzy dwoma punktami jest przydzielana losowo, lecz programista ma możliwość ustalenia maksymalnej różnicy pomiędzy oszacowaniem a rzeczywistą długością. Tak więc za miarę dokładności heurystyki można przyjąć maksymalną różnicę długości drogi oszacowanej i drogi rzeczywistej. Pomiarów dokonano

na próbie 10 000 węzłów grafu ze względu na długi czas wykonania algorytmu.



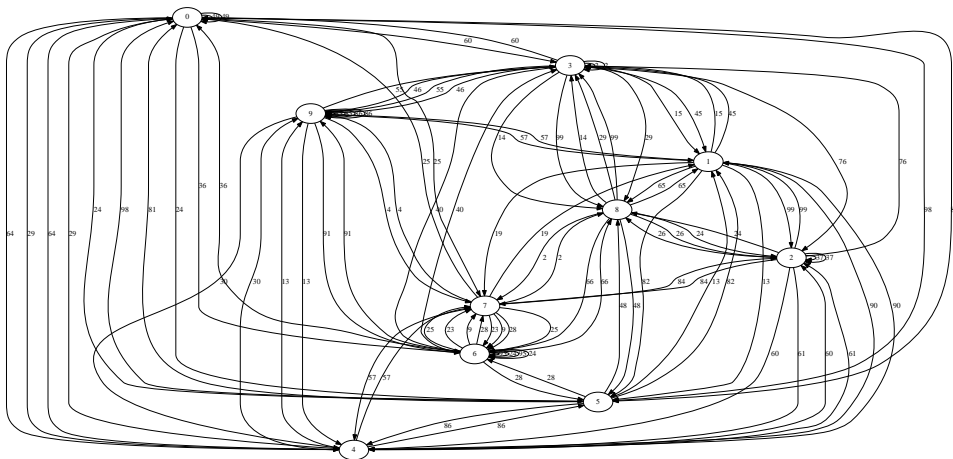
Rysunek 5: Szybkość wykonania algorytmu A*

Liczby na osi poziomej wykresu 5 to różnica pomiędzy rzeczywistą drogą a jej oszacowaniem zgodnie z zastosowaną metryką. Jak widać, im mniejsza jest ta różnica, tym szybciej pracuje algorytm, gdy długość ścieżki estymujemy z nadmiarem, czas jest również krótszy, jednak możemy nie znaleźć najkrótszej ścieżki. Powyższy wykres jest spójny z teorią, tzn. gdy heurystyka jest o wiele mniejsza od rzeczywistej długości, działanie algorytmu coraz bardziej zbliża się do algorytmu Dijkstry, w drugą stronę algorytm upodabnia się do tzw. best - first search.

3 Wnioski

- Wszystkie powyższe algorytmy osiągają podobną złożoność obliczeniową.
- wykorzystanie uprzednio stworzonych struktur danych, jak np. tablica asocjacyjna, pozwoliło na bardziej intuicyjną implementację algorytmów.
- Niestety badane algorytmy mogą okazać się mało efektywnym sposobem przeszukiwania grafów. Efekty ich działania są zbyt mocno zależne od struktury grafu. O efektywności wykonania algorytmu w wielu sytuacjach decyduje losowe wybranie szukanego elementu, co często może prowadzić do sytuacji optymistycznej, ale także pesymistycznej.

- Algorytm A* okazał się bardzo wydajnym algorytmem wyszukiwania najkrótszej ścieżki w grafie. Gdy potrafimy wyznaczyć dokładną heurystykę, algorytm łączy szybkość oraz możliwość optymalizacji (znalezienia najkrótszej ścieżki). Przyjęte oszacowanie ma wpływ na szybkość działania oraz na dokładność (dla przeszacowania algorytm może znaleźć inną ścieżkę).



Rysunek 6: Przykładowy rysunek grafu dla 10 węzłów. Graf wykoanany za pomocą programu GrapViz, plik źródłowy: graf.dot