Sprawozdanie z ćwiczenia laboratoryjnego

Tomasz Mikołajewski

2014

Spis treści

1 Wprowadzenie

Niniejszy dokument powstał w ramach przedmiotu Projektowanie Algorytmów i Metod Sztucznej Inteligencji. Jest on rezultatem przeprowadzonych ćwiczeń w laboratorium oraz pracy wykonanej w domu.

Ćwiczenie polegało na stworzeniu struktury grafu i przeprowadzeniu analizy złożoności algorytmu przeszukiwania. Podczas tekstu sprawdzano algorytmy depth first search (przeszukiwanie wszerz) i breadth first search (przeszukiwanie wgłąb). Podczas wykonywania funkcji przeszukującej mierzony był czas potrzebny do wykonania algorytmu. Testowanie powtarzano wielokrotnie, aby uzyskać ilość danych pozwalającą na wyznaczenie przybliżonej funkcji opisującej złożoność algorytmu.

2 Kod programu

Program został oparty o wcześniej stworzone pliki zawierające kod potrzebny do wykonania *benchmark-u* podprogramu, czyli określenia czasu jaki potrzebny jest na wykonanie konkretnego zadania. Główny program zawiera funkcję otwierającą pliki z danymi oraz zapisującą rezultat operacji do pliku. Zdecydowanie ułatwia to prace przy analizie zadanego algorytmu.

Algorytmy, które były testowane to przeszukiwanie:

- breadth first search
- depth first search

3 Pomiary

Pomiary zostały przeprowadzone na dużych plikach zmienne zawierające wierzchołki oraz wagi poszczególnych krawędzi. Każdy z algorytmów był testowany dla co najmniej 5 różnych rozmiarów problemu powtarzanych dla plików generowanych automatycznie. Aby otrzymać rzetelny pomiar, każdy z testów był przeprowadzony 10-krotnie. Oznacza to iż każdy z algorytmów był testowany co najmniej 50 razy.

Wykonanie pomiarów wymagało szczególnego doboru wielkości plików z danymi. Celem ćwiczenia było sprawdzenie złożoności algorytmu przeszukiwania. Warto w tym miejscu zauważyć, że przeszukiwanie tablicy z hashowaniem ma zmienna złożoność obliczeniową zależną od ilości elementów. Dla niewielkiej ilości elementów (mniejszej lub zbliżonej do ilości miejsc w tablicy wynoszącej podczas wykonywania ćwiczenia 15625 elementów), złożoność obliczeniowa wyszukiwania elementu w tabeli wynosi 1(N). Dla większych plików złożoność przechodzi stopniowo w zależność liniową. Wynika z przedstawionego rozumowania, że aby zbadać złożoność samego przeszukiwania należy korzystać z plików, w których ilość danych nie powoduje zmiany złożoności algorytmu. Jednocześnie ilość danych nie mogła być zbyt mała, ponieważ w wypadku plików z losowymi elementami(takimi jak zastosowane w ćwiczeniu) mogłoby dojść do sytuacji, w której struktura grafu jest bardzo podzielona. W tym wypadku przeszukiwanie przeprowadzałoby operacje na małych podgrafach, a z otrzymanych danych nie można by wnioskować o zło-

żoności obliczeniowej. Stosując się do wszystkich powyższych analiz przeprowadzono pomiary dla ilości danych zbliżonych oraz nieznacznie mniejszych od wielkości tablicy hashującej. Dodatkowo elementy odwiedzane przez funkcje wykonujące przeszukiwanie były zliczane, aby nie uwzględniać pozostałych podgrafów.

4 Wyniki pomiarów

Po przeprowadzeniu serii pomiarów otrzymano wyniki przedstawione w tabelach. Na podstawie wyników utworzono wykresy z naniesionymi funkcjami wzorcowymi. Na podstawie wykresów wyciągnięto wnioski.

4.1 Breadth first search

Jest to algorytm wykonujący przeszukiwanie zaczynając od sąsiadów najbliższych elementowi początkowemu. Wykorzystuje on listę typu FIFO, na której gromadzone są wierzchołki, do których należy się odwołać. Podczas wykonywania algorytmu zapisywana jest ódległość "pomiędzy punktem startowym, a dany elementem. Jest to przydatna informacja np: przy szukania najkrótszej drogi pomiędzy punktami.

Dane zebrane podczas pomiarów zostały przedstawione w Tablica 1. Na ich podstawie wykreślono Rysunek 1

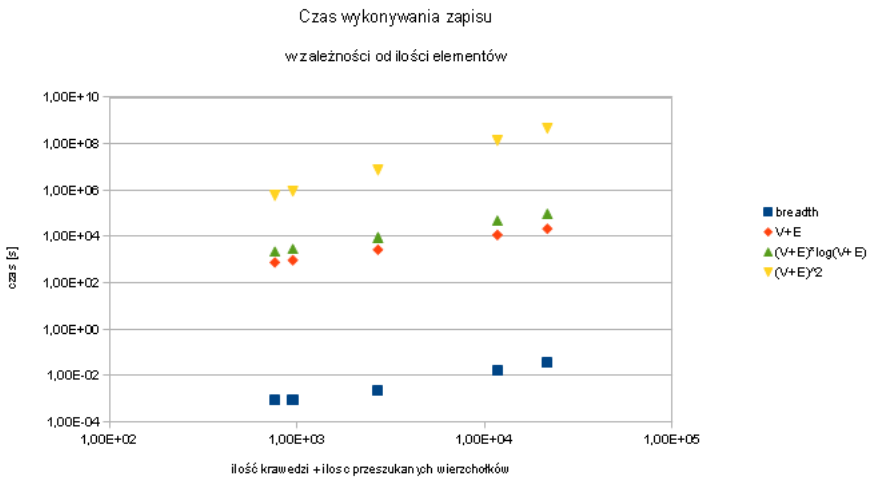
Tablica 1: Przeszukiwanie wszerz

	0,0007	0,0008	0,0158	0,0022	0,0342
	0,001	0,0009	0,016	0,0023	0,0362
	0,0012	0,0009	0,0161	0,0025	0,035
	0,0008	0,0008	0,0161	0,0027	0,0341
	0,0006	0,0009	0,0154	0,0016	0,0343
	0,0006	0,0014	0,0159	0,0023	0,0349
	0,0014	0,0006	0,0159	0,002	0,0357
	0,0014	0,0008	0,0159	0,0026	0,0357
	0,0006	0,0011	0,0155	0,0021	0,0366
	0,0006	0,0005	0,0162	0,0017	0,035
czas średni	0,00089	0,00087	0,01588	0,0022	0,03517
ilość elementów	310	250	3900	900	7200
ilość krawędzi	640	512	7800	1800	14400

4.2 Depth first search

Algorytm wykonujący przeszukiwanie w głąb. Polega on na wywoływaniu sąsiadów wierzchołków aż do momentu osiągnięcia krańca grafu lub powrotu do wcześniej odwiedzonego wierzchołka.

Teoretycznie powinien on mieć złożoność podobną do wcześniej omawianego algorytmu. Cześć algorytmu odpowiedzialna za wpisywanie odległości od punktu początkowego została przeniesiona do komentarza. Może to powodować widoczną różnicę pomiędzy algorytmami. Zebrane dane zostały przedstawione w Tablica 2, a następnie zwizualizowane na Rysunek 2



Rysunek 1: Przeszukiwanie wszerz

4.3 Dodatkowe porównania danych i wykresy

W ramach porównania algorytmów przedstawiono wszystkie dane na jednym Rysunku 3

Przedstawiono również omawianą wcześniej zależność pomiędzy zwiększającą się ilością argumentów, a zmianą złożoności obliczeniowej. Na Rysunku 4 widać już wyraźnie, że wykonywany algorytm posiada złożoność kwadratową co jest związane ze zwiększającą się złożonością wyszukiwania przy dużej ilości danych.

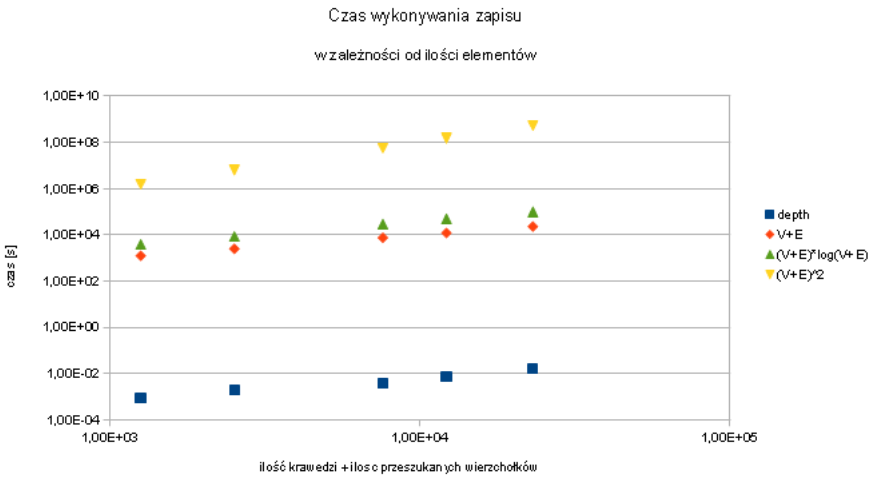
Tablica 2: Przeszukiwanie w głąb

	0,00076	0,0159	0,0078	0,00369	0,00077
	0,00107	0,01592	0,00779	0,00433	0,00094
	0,00094	0,01622	0,0075	0,0039	0,00125
	0,00079	0,01573	0,00705	0,00358	0,00094
	0,00094	0,01592	0,00719	0,00358	0,00141
	0,00047	0,01666	0,00736	0,00357	0,00142
	0,00109	0,01543	0,0072	0,00357	0,00141
	0,00048	0,01484	0,0075	0,00389	0,00109
	0,0011	0,01642	0,0069	0,00406	0,00108
	0,00094	0,0156	0,00732	0,0038	0,0094
czas średni	0,000858	0,015864	0,007361	0,003797	0,001971
ilość elementów	280	5050	2800	1800	575
ilość krawędzi	980	18100	9380	5800	1949

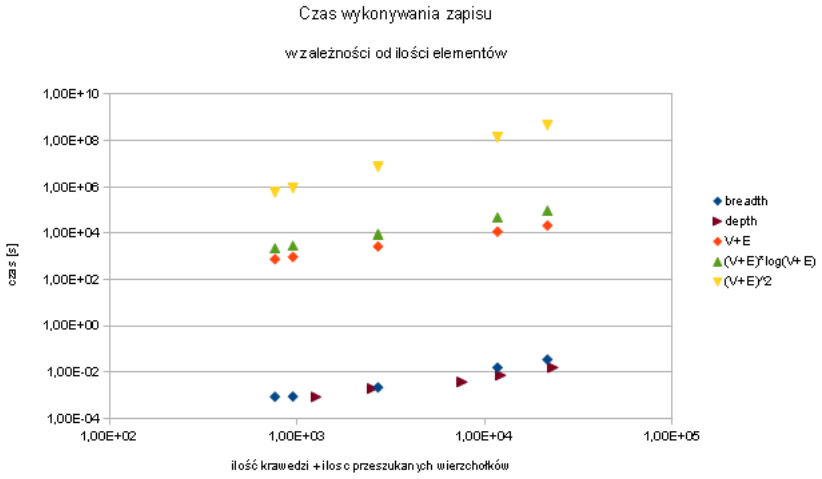
5 Wnioski

Na podstawie danych i wykresów można wyciągnąć następujące wnioski:

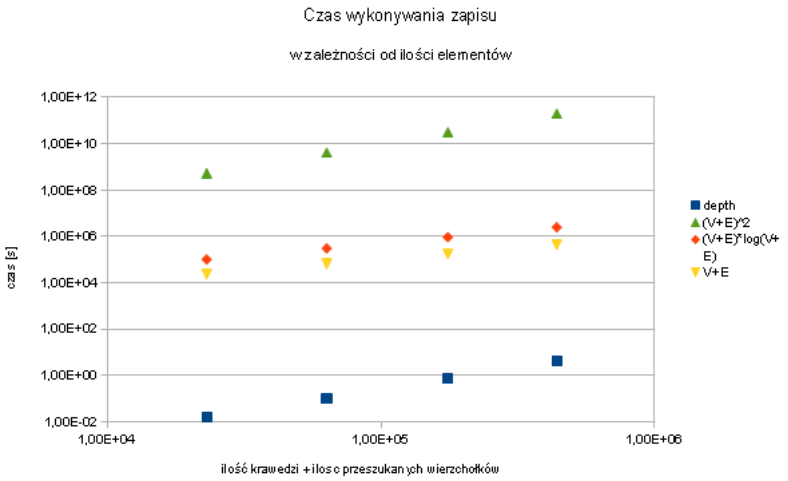
- oba przedstawione algorytmy posiadają podobną złożoność obliczeniową
- przy małych ilościach danych algorytmy miały złożoność liniową. Można przypuszczać, że złożoność ta wynosiła około E + N. Taka złożoność wynika z rozważań teoretycznych, ponieważ w obu algorytmach powinien zostać odwiedzony każdy z wierzchołków i każda krawędź.
- oba algorytmy przeszukiwały cały podgraf i zwracały tą samą liczbę świadczącą o ilości odwiedzonych elementów
- zauważono, że częściej spotykanym rezultatem losowania krawędzi jest powstanie dużego podgrafu i małych-kilkuelementowych podgrafów (nigdy nie wystąpiła sytuacja w której powstały dwa (lub więcej) grafy o podobnej (dużej) ilości elementów). Jest to całkowicie zrozumiałe, gdy zastanowimy się nad prawdopodobieństwem połączenia się dwóch dużych grafów (o wielu elementach) i przyrównamy je do prawdopodobieństwa połączenia się małego podgrafu z dużym.
- przy dużej ilości danych algorytmy przyjmują złożoność kwadratową, ponieważ wyszukiwanie elementów przyjmuje złożoność liniową



Rysunek 2: Przeszukiwanie w głąb



Rysunek 3: Porównanie wykresów



Rysunek 4: Zmiana złożoności