

# Systemy Operacyjne: Projekt

Porównanie prędkości dwukierunkowego algorytmu Dijkstry  
wykorzystującego jeden lub dwa wątki.

Sprawozdanie

Aleksander Dziągwa  
281055

## 1 Wprowadzenie

- 1.1 W tym projekcie zaimplementowałem dwukierunkowy algorytm Dijkstry wykorzystujący dwie kolejki priorytetowe. Głównym celem projektu było porównanie wydajności algorytmu działającego na jednym lub dwóch wątkach.
- 1.2 Całość projektu została napisana w języku C, do zarządzania wielowątkowością wykorzystałem bibliotekę *OpenMP* w wersji 4.5. Całość projektu dostępna jest na moim [githubie](#).
- 1.3 Do kompilacji projektu użyłem systemu *CMake*. Przy kompilacji wersji ostatecznej użyłem flag optymalizacyjnych: *-Ofast -march=native -flto -fomit-frame-pointer*.

## 2 Sposób wykonania pomiarów

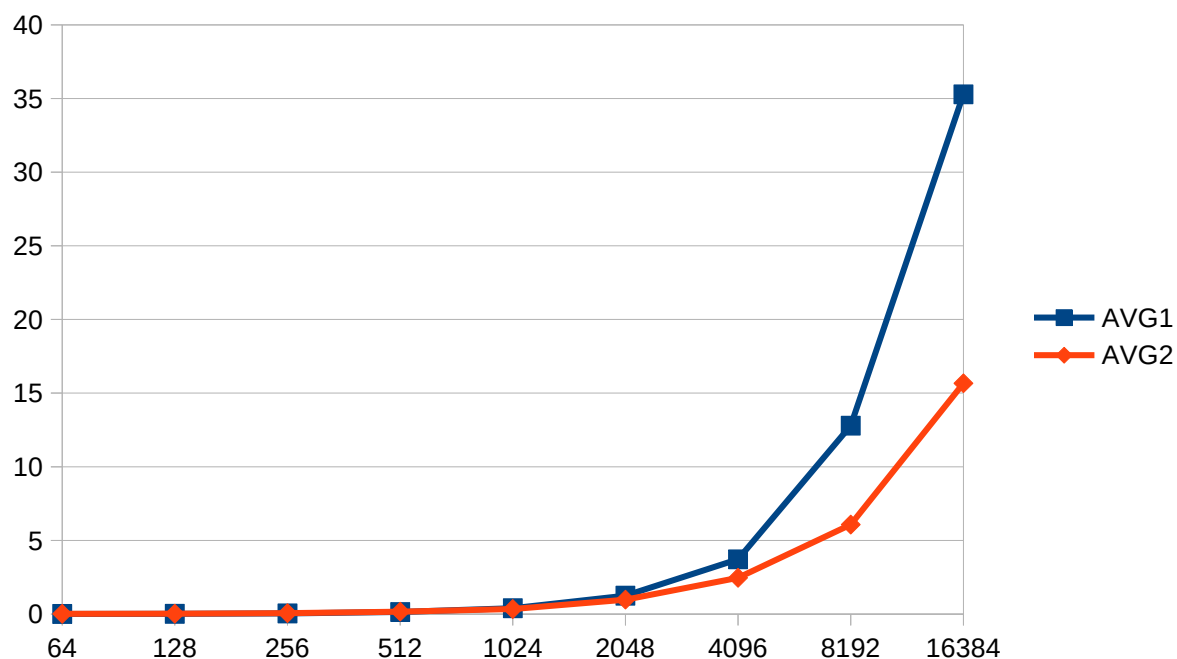
- 2.1 Pomiary prędkości wykonania się algorytmu Dijkstry wykonałem na 9 rozmiarach grafów. Dla każdego rozmiaru sprawdziłem 3 stopnie gęstości grafu, tj. 25%, 50% oraz 99%.
- 2.2 Dane wykorzystane w obliczeniach zostały wygenerowane losowo za pomocą funkcji *rand\_r()*.
- 2.3 Do pomiaru czasu wykonania wykorzystałem funkcję *omp\_get\_wtime()* z biblioteki *OpenMP*. Pomiar rozpoczyna się bezpośrednio przed wywołaniem funkcji i kończy zaraz po zakończeniu wykonywania funkcji.
- 2.4 Obydwa algorytmy dostają jako wejście ten sam zestaw danych, po skończeniu obliczeń przez obydwa program porównuje uzyskane wyniki. Jeżeli nie są one identyczne oznacza to, że doszło do błędu podczas obliczeń. Błędny wynik jest odrzucany a obliczenia są powtarzane na innym zestawie danych.

### 3 Wyniki oraz ich analiza

3.1 Wyniki uzyskane dla grafów o gęstości 25%. AVG1 oraz AVG2 to odpowiednio średnie czasy wykonania się algorytmu na jednym i dwóch wątkach. ODCH1 i ODCH2 to wartości odchylenia standardowego uzyskanych wyników.

Tabela 1. Wyniki pomiarów dla grafów o gęstości 25%

Ilość wierzchołków	AVG1 (ms)	ODCH1 (ms)	AVG2 (ms)	ODCH2 (ms)
64	0.01	0.01	0.02	0.01
128	0.02	0.01	0.03	0.02
256	0.05	0.02	0.07	0.03
512	0.14	0.07	0.17	0.06
1024	0.4	0.24	0.34	0.09
2048	1.25	0.64	0.98	0.3
4096	3.72	1.9	2.46	0.7
8192	12.79	7.32	6.07	2.47
16384	35.28	24.32	15.67	7.37



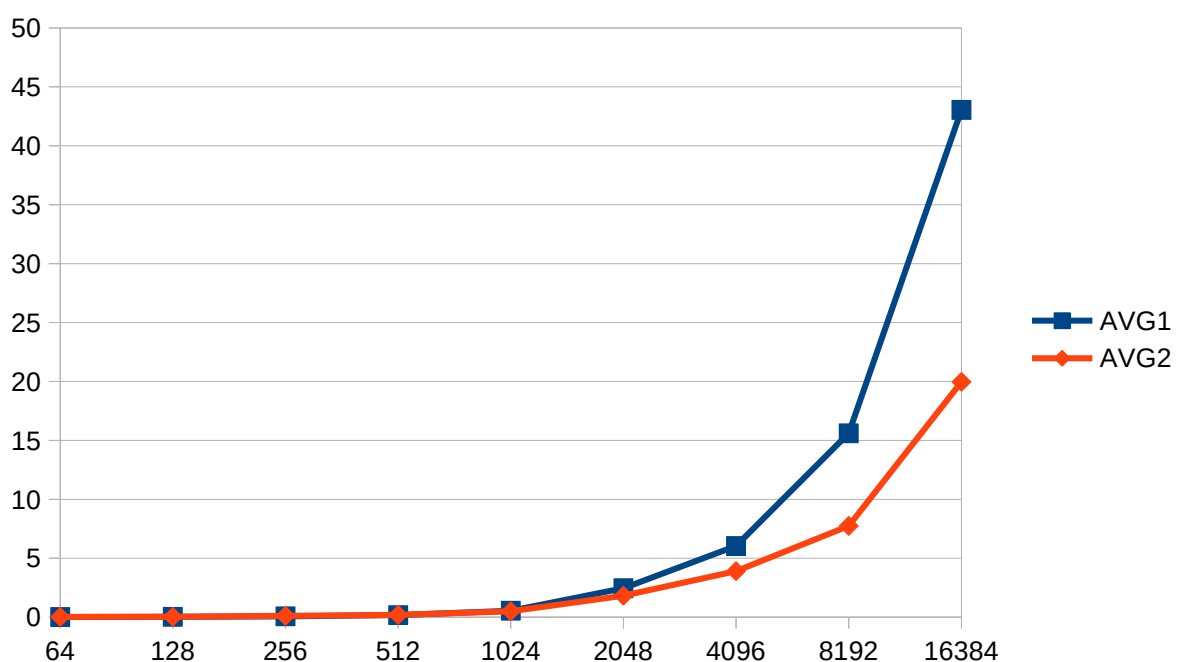
Wykres 1. Zależność czasu wykonania się algorytmu od rozmiaru grafu (25%)

Jak widać na powyższym wykresie czasy wykonania się algorytmów dla niewielkich grafów są bardzo zbliżone. Dopiero od około 8 tysięcy wierzchołków można zobaczyć spodziewane, dwukrotne przyspieszenie algorytmu. Ciekawostką wartą odnotowania jest również zdecydowanie niższa wartość odchylenia standardowego dla algorytmu wykorzystującego dwa wątki.

3.2 Wyniki uzyskane dla grafów o gęstości 50%. AVG1 oraz AVG2 to odpowiednio średnie czasy wykonania się algorytmu na jednym i dwóch wątkach. ODC1 i ODC2 to wartości odchylenia standardowego uzyskanych wyników.

Tabela 2. Wyniki pomiarów dla grafów o gęstości 50%

Ilość wierzchołków	AVG1 (ms)	ODCH1 (ms)	AVG2 (ms)	ODCH2 (ms)
64	0.011	0.006	0.021	0.009
128	0.027	0.013	0.033	0.017
256	0.073	0.035	0.114	0.058
512	0.178	0.081	0.195	0.057
1024	0.549	0.246	0.495	0.18
2048	2.462	1.299	1.821	0.336
4096	6.036	3.495	3.91	0.954
8192	15.606	8.745	7.738	3.387
16384	43.044	29.666	19.971	9.396



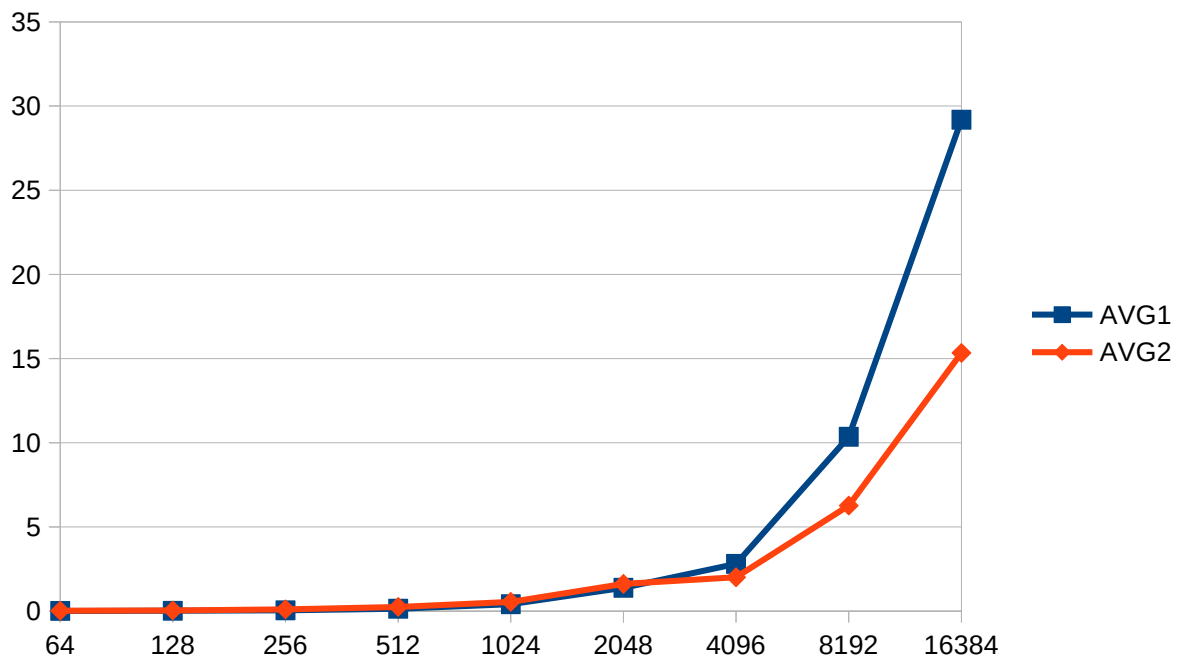
Wykres 2. Zależność czasu wykonania się algorytmu od rozmiaru grafu (50%)

Ponownie minimalne zyski na czasie wykonania możemy zaobserwować przy grafach o wielkości około 2 tysięcy wierzchołków i dwukrotny spadek czasu wykonania od około 8 tysięcy wierzchołków.

3.3 Wyniki uzyskane dla grafów o gęstości 50%. AVG1 oraz AVG2 to odpowiednio średnie czasy wykonania się algorytmu na jednym i dwóch wątkach. ODCH1 i ODCH2 to wartości odchylenia standardowego uzyskanych wyników.

Tabela 3. Wyniki pomiarów dla grafów o gęstości 99%

Ilość wierzchołków	AVG1 (ms)	ODCH1 (ms)	AVG2 (ms)	ODCH2 (ms)
64	0.009	0.004	0.023	0.011
128	0.019	0.012	0.042	0.021
256	0.048	0.015	0.102	0.043
512	0.141	0.071	0.247	0.059
1024	0.409	0.12	0.551	0.113
2048	1.385	0.558	1.616	0.329
4096	2.804	1.98	2.012	0.441
8192	10.355	5.057	6.271	1.91
16384	29.188	22.889	15.335	6.307



Wykres 3. Zależność czasu wykonania się algorytmu od rozmiaru grafu (99%)

I w tym przypadku obserwujemy zdecydowane przyspieszenie algorytmu dwuwątkowego zaczynając od grafów o rozmiarze około 8 tysięcy wierzchołków. Warto zauważyć, że odwrotnie niż poprzednio, algorytm dwuwątkowy jest wolniejszy dla grafów o rozmiarze 2048 krawędzi, choć poprzednio był szybszy. Ponownie wartość odchylenia standardowego dla algorytmu wykorzystującego dwa wątki była zdecydowanie niższa.

## 4 Wnioski

- 4.1 Algorytm dwukierunkowy Dijkstry wykorzystujący dwa wątki jest szybszy od wykorzystującego jeden wątek dla dużych grafów (powyżej 8 tysięcy wierzchołków). Przy mniejszych grafach algorytm jednowątkowy jest delikatnie szybszy z powodu dodatkowych instrukcji narzuconych przez rozdzielenie pracy na dwa wątki.
- 4.2 Odchylenie standardowe czasów wykonania się algorytmu dwuwątkowego jest zdecydowanie niższe od algorytmu Dijkstry na jednym wątku.
- 4.3 Gęstość grafu nie ma wpływu na czas wykonywania się algorytmu. Czasy obliczeń dla grafów o gęstości 25%, 50% oraz 99% były bardzo zbliżone.
- 4.4 Implementacja dwukierunkowego algorytmu Dijkstry z użyciem dwóch wątków przynosi wymierne korzyści wydajnościowe dla dużych grafów, przy jednoczesnej poprawie stabilności pomiarów. Dla mniejszych instancji wersja jednowątkowa jest zbliżona wydajnościowo.