



Algorytmy grafowe

SORTOWANIE TOPOLOGICZNE DFS I BFS
PAWEŁ KOLEC

Wstęp

Sprawozdanie porusza zagadnienie sortowania topologicznego wierzchołków grafu. Skupia się na dwóch metodach: procedurze przechodzenia w głąb (DFS) oraz wszerz (BFS). Analizowane są również dwie reprezentacje maszynowe grafu skierowanego: macierz sąsiedztwa i lista następników. Celem jest przedstawienie zasad działania, implementacji, złożoności czasowej i przestrzennej, a także zalet i ograniczeń tych metod.

W pierwszej części sprawozdania wykorzystałem dwie popularne reprezentacje maszynowe grafu: macierz sąsiedztwa i listę następników.

1. Macierz sąsiedztwa:

Macierz sąsiedztwa to dwuwymiarowa tablica o rozmiarze $n \times n$, gdzie n oznacza liczbę wierzchołków w grafie. W macierzy sąsiedztwa wartość na pozycji (i, j) wskazuje, czy istnieje krawędź między wierzchołkami i oraz j . Jeśli istnieje krawędź, to wartość ta może przechowywać dodatkowe informacje, takie jak waga krawędzi.

Złożoność pamięciowa:

Macierz sąsiedztwa ma złożoność pamięciową $O(n^2)$, ponieważ przechowuje informacje o każdej możliwej parze wierzchołków.

Złożoność operacyjna:

Znalezienie pojedynczej krawędzi: $O(1)$, ponieważ wystarczy odczytać wartość na odpowiedniej pozycji w macierzy.

Znalezienie wszystkich następników wierzchołka: $O(n)$, ponieważ trzeba przejrzeć cały wiersz odpowiadający danemu wierzchołkowi.

2. Lista następników:

Lista następników to struktura danych, w której dla każdego wierzchołka przechowywana jest lista jego bezpośrednich następników (czyli wierzchołków, do których istnieje krawędź). Dodatkowo, można przechowywać informacje o wagach krawędzi.

Złożoność pamięciowa:

Lista następników ma złożoność pamięciową $O(n + m)$, gdzie n to liczba wierzchołków, a m to liczba krawędzi. Przechowuje się listę następników dla każdego wierzchołka oraz dodatkowe informacje o krawędziach.

Złożoność operacyjna:

Znalezienie następnika wierzchołka v : $O(n)$, ponieważ należy przejrzeć listę następników wierzchołka v .

Znalezienie wszystkich następników v : $O(n)$, ponieważ trzeba przejrzeć całą listę następników wierzchołka v .

Sortowanie topologiczne DFS:

Wstęp:

Sortowanie topologiczne jest algorytmem stosowanym w teorii grafów, który pozwala na uporządkowanie wierzchołków w acyklicznym grafie skierowanym. Jest szczególnie przydatne w sytuacjach, gdzie istnieją zależności pomiędzy elementami, które muszą być spełnione w określonej kolejności.

Jak działa:

Sortowanie topologiczne metodą Depth-First Search (DFS) polega na przeszukiwaniu grafu w głąb, zaczynając od dowolnego wierzchołka, a następnie odwiedzając rekurencyjnie wszystkie nieodwiedzone sąsiednie wierzchołki. Po zakończeniu rekurencyjnego przeglądu wszystkich sąsiadów danego wierzchołka, dodajemy go do listy wynikowej.

Złożoności:

Dla macierzy sąsiedztwa:

Złożoność czasowa: $O(V^2)$ - czas potrzebny na znalezienie wszystkich następników dla danego wierzchołka wynosi $O(V)$, ale musimy to zrobić dla każdego z V wierzchołków. Dla wszystkich krawędzi musimy wykonać operację, więc złożoność wynosi $O(V^2)$.

Złożoność pamięciowa: $O(V^2)$ - macierz sąsiedztwa wymaga pamięci na przechowywanie informacji o wszystkich możliwych połączeniach pomiędzy wierzchołkami, co daje złożoność kwadratową względem liczby wierzchołków.

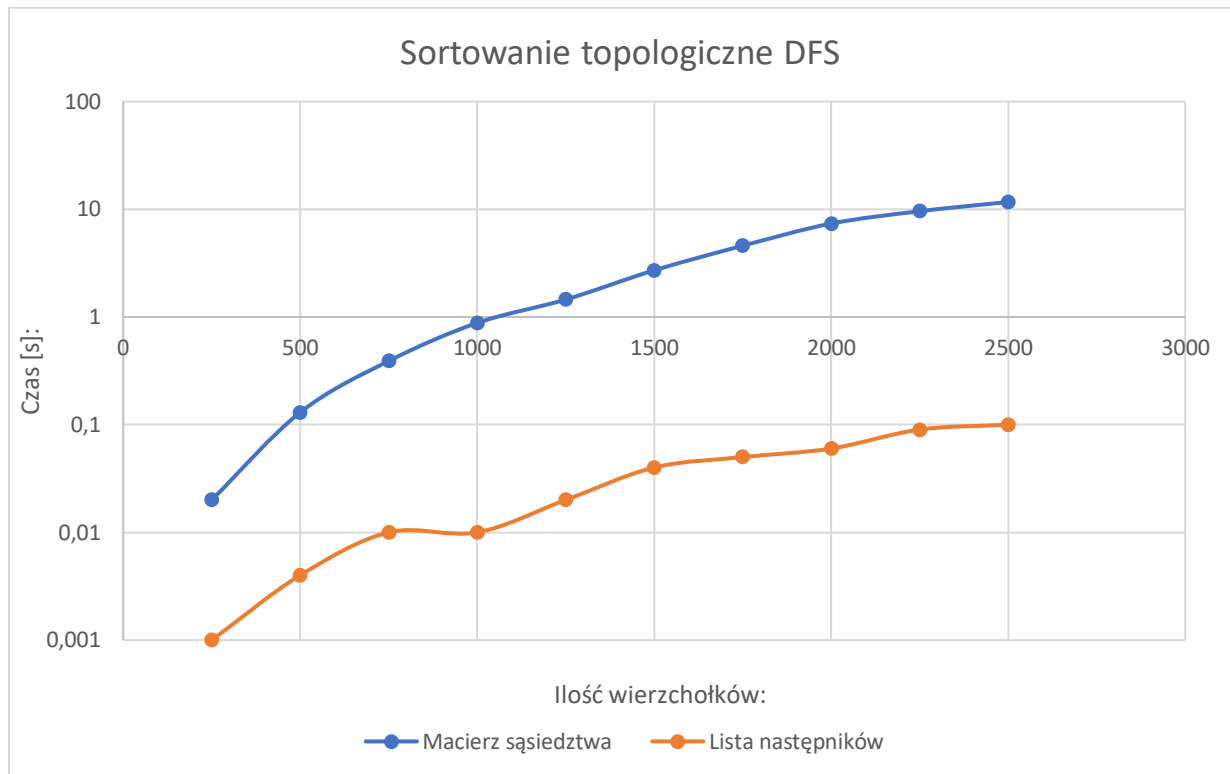
Dla listy następników:

Złożoność czasowa: $O(V + E)$ - czas potrzebny na znalezienie wszystkich następników dla wszystkich wierzchołków wynosi $O(V)$. Dodatkowo, musimy przejrzeć wszystkie krawędzie, co zajmuje czas $O(E)$.

Złożoność pamięciowa: $O(V + E)$ - lista następników wymaga pamięci na przechowywanie informacji o każdym wierzchołku oraz jego listy następników, co daje złożoność liniową względem liczby wierzchołków i krawędzi.

Podsumowując, dla sortowania topologicznego DFS lista następników ma złożoność czasową $O(V + E)$, co jest bardziej efektywne niż macierz sąsiedztwa o złożoności $O(V^2)$. Złożoność pamięciowa jest również niższa dla listy następników ($O(V + E)$) w porównaniu do macierzy sąsiedztwa ($O(V^2)$). Dlatego lista następników jest lepszym sposobem reprezentacji grafu przy implementacji sortowania topologicznego DFS.

Zależność czasu obliczeń t od liczby n wierzchołków w sortowaniu topologicznym DFS dla różnych reprezentacji maszynowych grafu:



Wnioski:

Reprezentacja grafu ma istotne znaczenie dla złożoności czasowej i pamięciowej algorytmu sortowania topologicznego DFS. W porównaniu między macierzą sąsiedztwa a listą następników, lista następników okazuje się lepszą reprezentacją pod względem efektywności.

Lista następników jest preferowaną reprezentacją grafu do sortowania topologicznego DFS ze względu na jej lepszą złożoność czasową i pamięciową w porównaniu do macierzy sąsiedztwa. Lista następników jest szczególnie przydatna w przypadku dużych grafów, gdzie efektywność obliczeniowa i zużycie pamięci są ważne. Wybór odpowiedniej reprezentacji grafu ma znaczący wpływ na wydajność i skuteczność algorytmu sortowania topologicznego DFS.

Sortowanie topologiczne BFS:

Jak działa:

Sortowanie topologiczne BFS rozpoczyna się od wybranego wierzchołka źródłowego. Najpierw odwiedza się wszystkich jego sąsiadów, a następnie przechodzi się do kolejnych wierzchołków, które nie zostały jeszcze odwiedzone. W trakcie przeglądania sąsiadów każdego wierzchołka, zlicza się stopnie wejściowe dla wszystkich wierzchołków. Stopień wejściowy wierzchołka oznacza liczbę krawędzi wchodzących do danego wierzchołka. W wyniku sortowania topologicznego BFS wierzchołki zostają uporządkowane w taki sposób, że żaden wierzchołek nie ma krawędzi skierowanej do wierzchołka wcześniejszego w kolejności.

Złożoności:

Porównując złożoności czasowe i pamięciowe dla macierzy sąsiedztwa i listy następników w kontekście sortowania topologicznego, można zauważyć następujące różnice:

Dla macierzy sąsiedztwa:

Czasowa: $O(V^2)$, gdzie V to liczba wierzchołków. Musimy przejrzeć całą macierz, aby znaleźć wszystkie następniki dla każdego wierzchołka. To prowadzi do złożoności kwadratowej, która staje się problematyczna dla dużych grafów.

Pamięciowa: $O(V^2)$, gdzie V to liczba wierzchołków. Musimy przechowywać macierz o rozmiarze $V \times V$, co wymaga pamięci proporcjonalnej do kwadratu liczby wierzchołków. Jest to szczególnie istotne dla dużych grafów, gdzie zużycie pamięci może być znaczące.

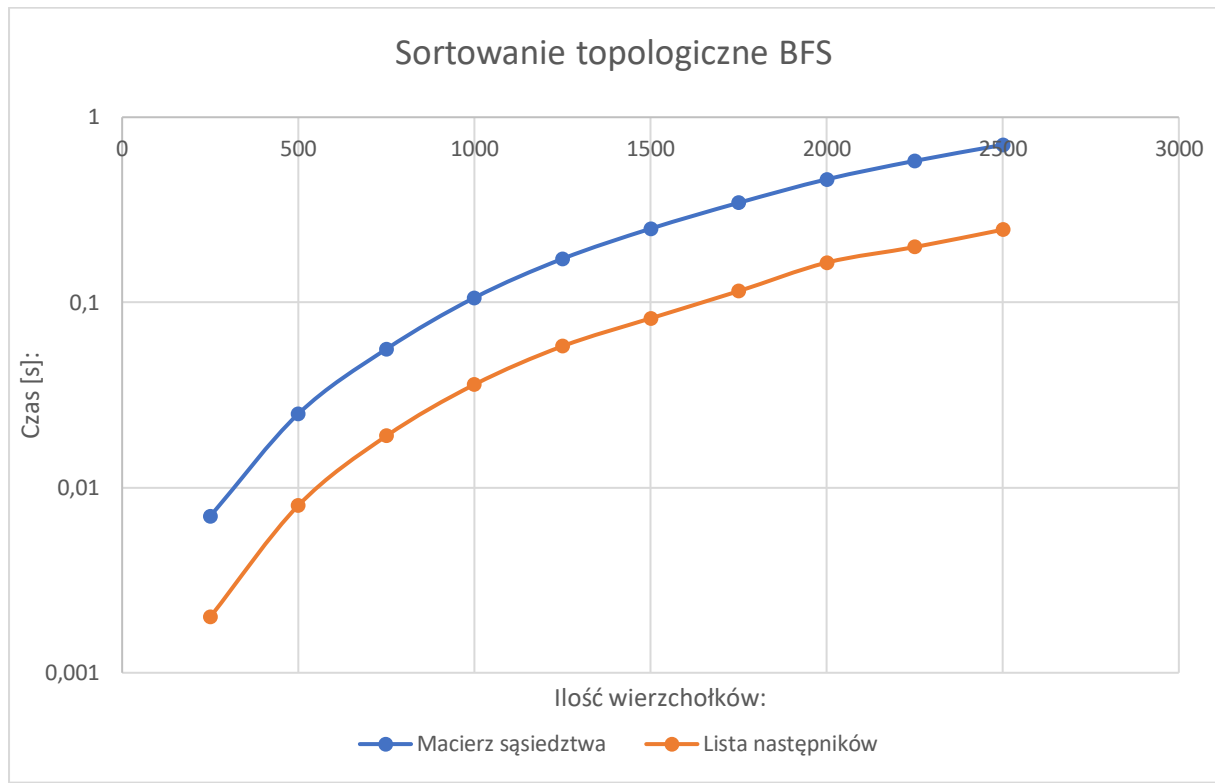
Dla listy następników:

Czasowa: $O(V + E)$, gdzie V to liczba wierzchołków, a E to liczba krawędzi. Przechodzimy przez wszystkie wierzchołki i krawędzie, co zajmuje czas proporcjonalny do ich sumy. Ta liniowa złożoność czasowa jest bardziej efektywna dla dużych grafów.

Pamięciowa: $O(V + E)$, gdzie V to liczba wierzchołków, a E to liczba krawędzi. Przechowujemy listy następników dla każdego wierzchołka, co zajmuje pamięć proporcjonalną do sumy liczby wierzchołków i krawędzi. Ta liniowa złożoność pamięciowa jest bardziej optymalna dla dużych grafów.

Podsumowując, lista następników jest lepszą reprezentacją grafu pod względem zarówno złożoności czasowej, jak i pamięciowej dla sortowania topologicznego. Zapewnia ona liniową złożoność czasową i pamięciową ($O(V + E)$), co jest bardziej efektywne dla dużych grafów. Macierz sąsiedztwa ma złożoność kwadratową zarówno czasową, jak i pamięciową ($O(V^2)$), co staje się problematyczne dla większych grafów. Dlatego lista następników jest preferowaną reprezentacją grafu przy implementacji sortowania topologicznego BFS.

Zależność czasu obliczeń t od liczby n wierzchołków w sortowaniu topologicznym BFS dla różnych reprezentacji maszynowych grafu:

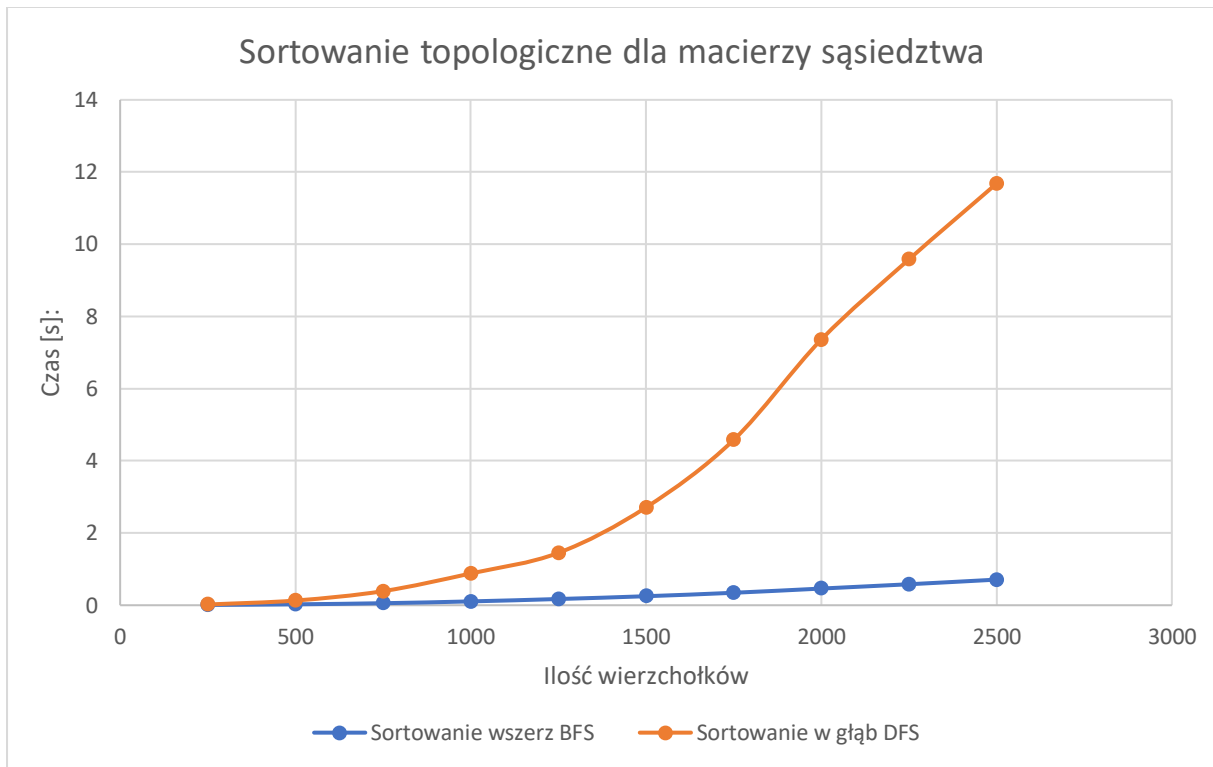


Wnioski:

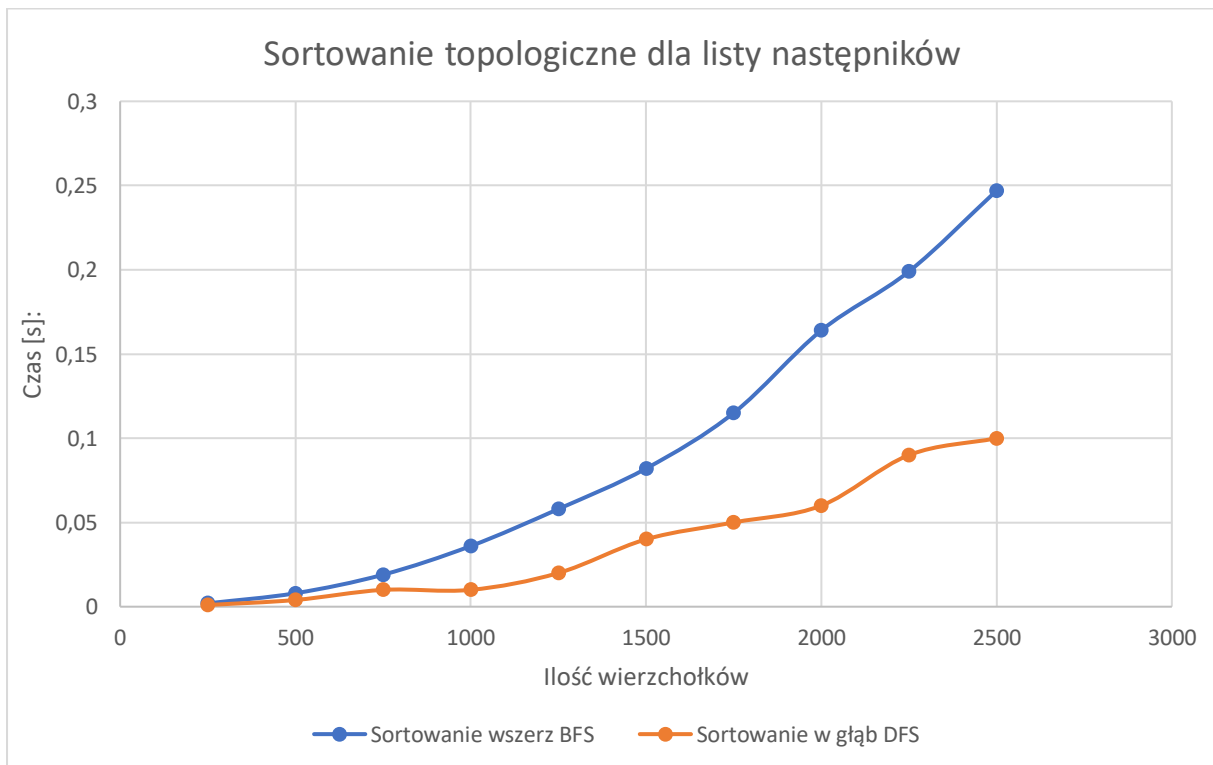
Reprezentacja grafu ma kluczowe znaczenie dla złożoności czasowej i pamięciowej algorytmu sortowania topologicznego BFS. W porównaniu między macierzą sąsiedztwa a listą następników, lista następników jest preferowaną reprezentacją, ze względu na jej lepszą efektywność zarówno czasową, jak i pamięciową. Jest szczególnie korzystna dla dużych grafów, gdzie optymalne wykorzystanie zasobów obliczeniowych i pamięci jest istotne. Wybór właściwej reprezentacji grafu ma istotny wpływ na wydajność i skuteczność algorytmu sortowania topologicznego BFS.

Porównanie metod sortowania topologicznego w zależności od reprezentacji maszynowej grafu:

Macierz sąsiedztwa:



Lista następników:



Wnioski:

Porównując metody sortowania topologicznego w zależności od reprezentacji grafu, lista następników jest lepszą opcją niż macierz sąsiedztwa. Dla macierzy sąsiedztwa, BFS działa lepiej niż DFS ze względu na złożoność czasową $O(V^2)$. Dla listy następników, DFS jest bardziej efektywny niż BFS ze względu na złożoność czasową $O(V + E)$. Ogólnie rzecz biorąc, lista następników jest bardziej efektywną reprezentacją grafu dla sortowania topologicznego zarówno przy użyciu DFS, jak i BFS, ze względu na lepszą złożoność czasową i pamięciową.