

Sprawozdanie – Algorytmy grafowe

Kacper Leporowski

Adam Malinowski

Sprawozdanie to ma na celu zaprezentowanie złożoności wybranych algorytmów działających na grafach. Algorytmy opisane w sprawozdaniu to algorytmy przechodzenia przez graf wszerz (breadth first search) i w głąb (depth first search), a także algorytm sortowania topologicznego w dwóch wersjach – oparty na algorytmach BFS i DFS. Do wykonania testów wygenerowane zostały skierowane acykliczne grafy (Directed acyclic graph, DAG) o nasyceniu łukami 50% o liczebności wierzchołków od 100 do 1000. Ten konkretny typ grafu został wybrany dlatego, że wyłącznie acykliczne skierowane grafy można sortować topologicznie.

Reprezentacje grafów, które zostały wybrane to macierz sąsiedztwa (Adjacency matrix, AM), lista następników (Adjacency list, AL) oraz tabela krawędzi (Edge list, EL).

Macierz sąsiedztwa

Złożoność pamięciowa tej reprezentacji grafu wynosi $O(V^2)$, gdzie V oznacza liczbę wierzchołków grafu. Znalezienie danej krawędzi w grafie ma złożoność $O(1)$, a znalezienie wszystkich następników wierzchołka – $O(V)$.

Złożoność algorytmów przechodzenia przez graf w postaci macierzy sąsiedztwa mają złożoność $O(V^2)$ – składa się ona z V wierszy oznaczających wierzchołki, a znajdowanie następników ma złożoność V . Podobnie sprawa ma się z algorytmami sortowania topologicznego, które oparte są na algorytmach BFS i DFS.

Lista następników

Złożoność pamięciowa AL to $O(V+E)$ – E oznacza liczbę krawędzi. Złożoność znalezienia krawędzi, a także znalezienia wszystkich następników wierzchołka jest taka sama i zależy od długości pojedynczej listy następników dla danego wierzchołka – w najgorszym przypadku jest to $O(V)$, a w średnim – $O(V/E)$.

Złożoność algorytmów grafowych w liście następników to $O(V+E)$ – lista każdego wierzchołka jest odwiedzana raz, a suma wszystkich list daje liczbę krawędzi, czyli E .

Tabela krawędzi

Złożoność pamięciowa tabeli krawędzi wynosi $O(2 \cdot E)$. Znalezienie danej krawędzi ma złożoność $O(E)$, podobnie jak znalezienie wszystkich następników wierzchołka.

Algorytmy grafowe wykonują się w czasie $V \cdot E$ – aby przejść po wszystkich krawędziach wychodzących z danego wierzchołka trzeba przejść przez całą listę, której długość wynosi E .

Ponieważ liczba krawędzi jest dużo większa od liczby wierzchołków, algorytmy wykonywane na tabeli krawędzi wykonują się znacznie dłużej, niż na AM i AL. Z tego powodu dla każdego algorytmu pokazane będą dwa wykresy – jeden z wszystkimi algorytmami, drugi bez tabeli krawędzi.

Przeszukiwanie wszerz

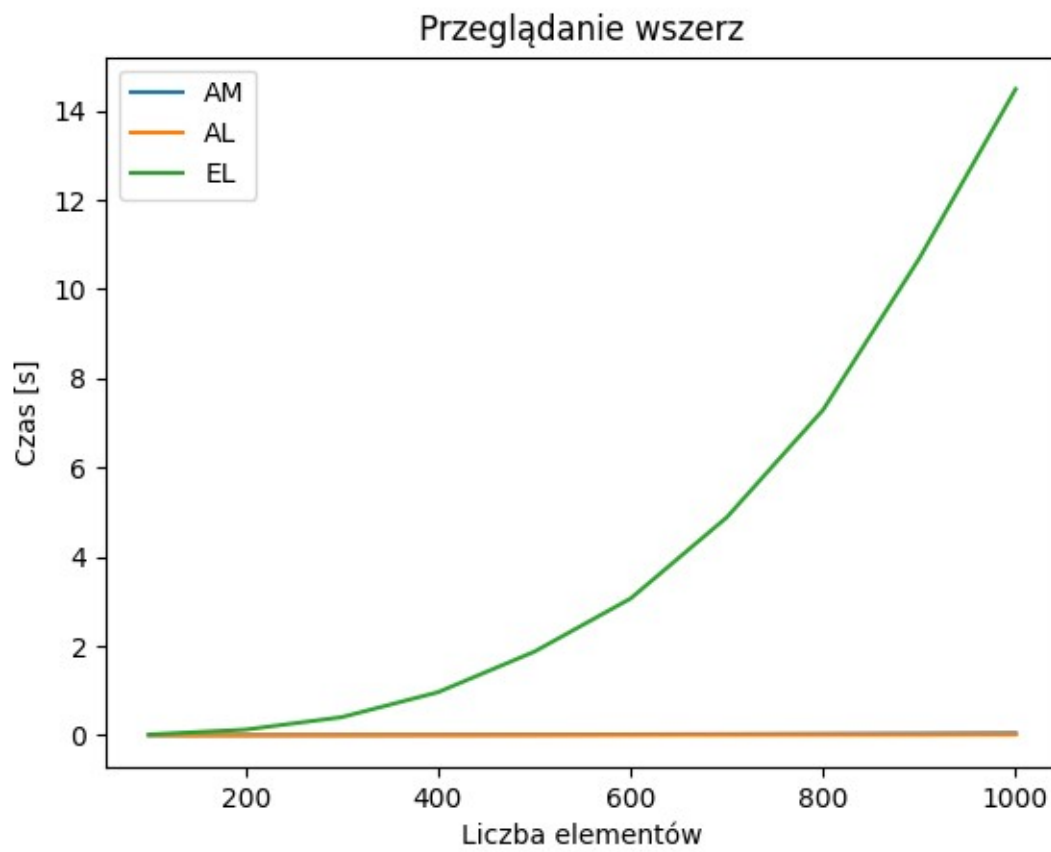


Figura 1: BFS - wszystkie algorytmy

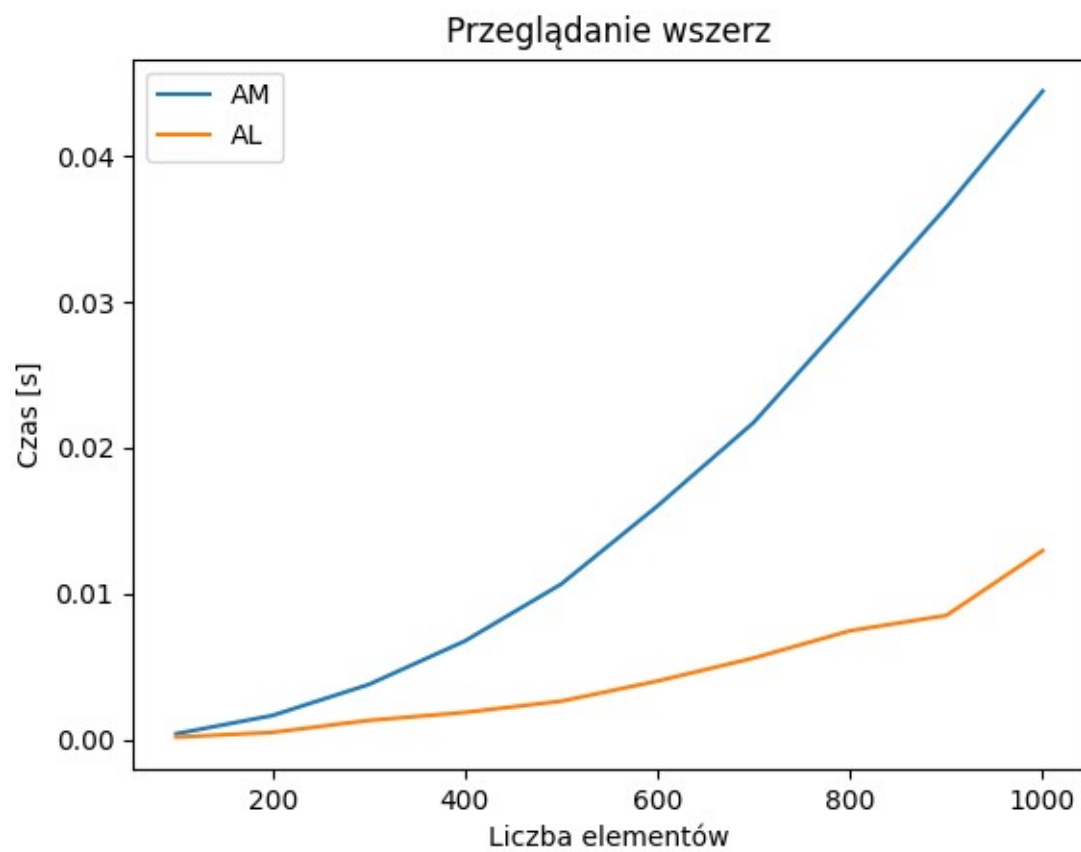


Figura 2: BFS - bez EL

Przeszukiwanie w głąb

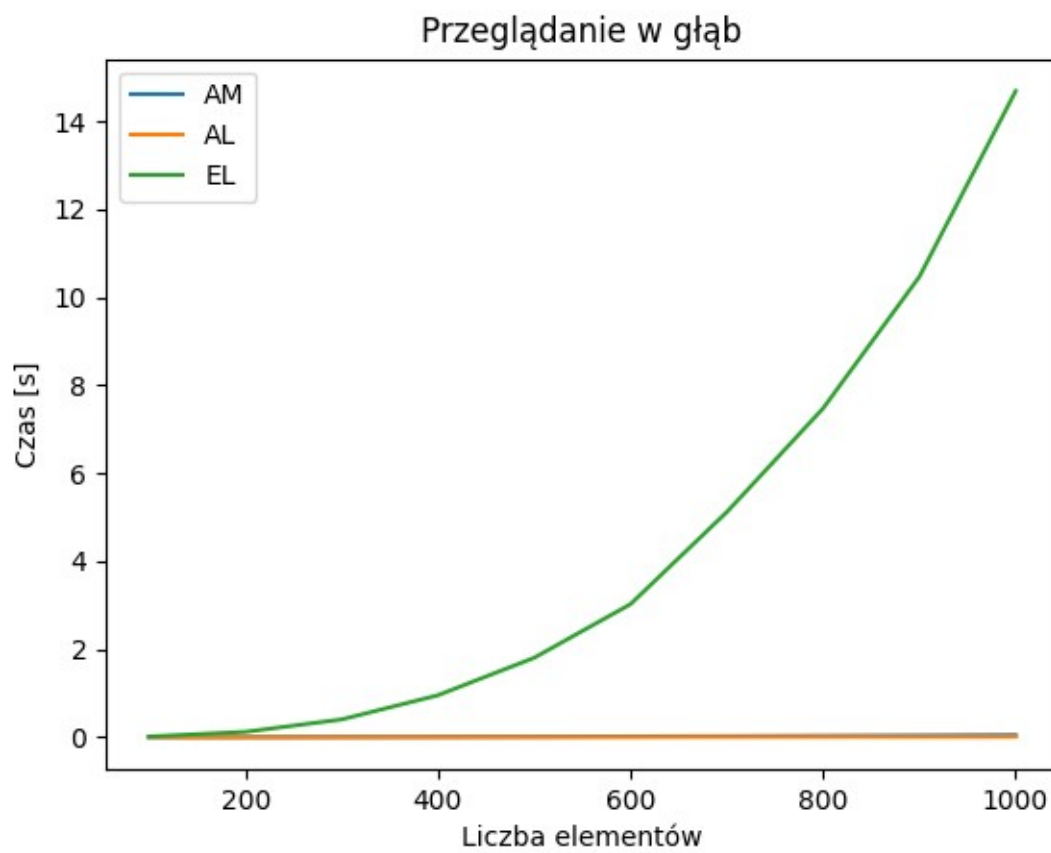


Figura 3: DFS - wszystkie algorytmy

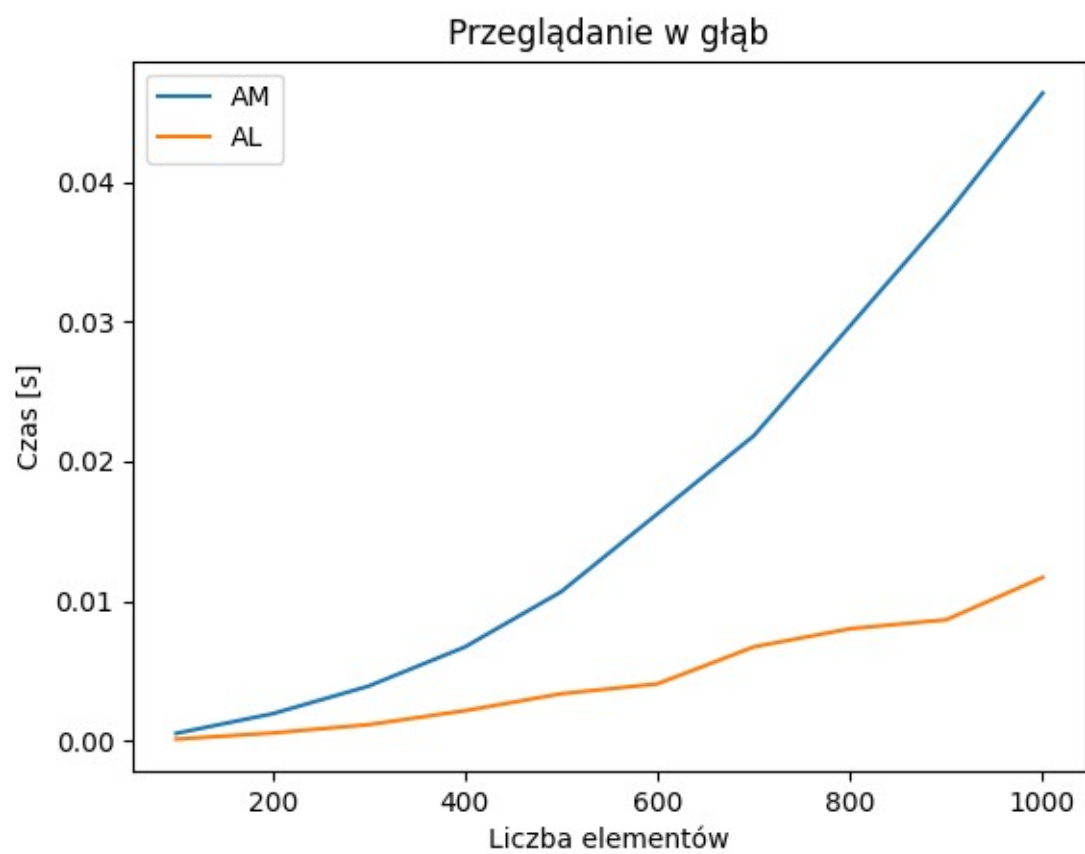


Figura 4: DFS - bez EL

Sortowanie topologiczne BFS

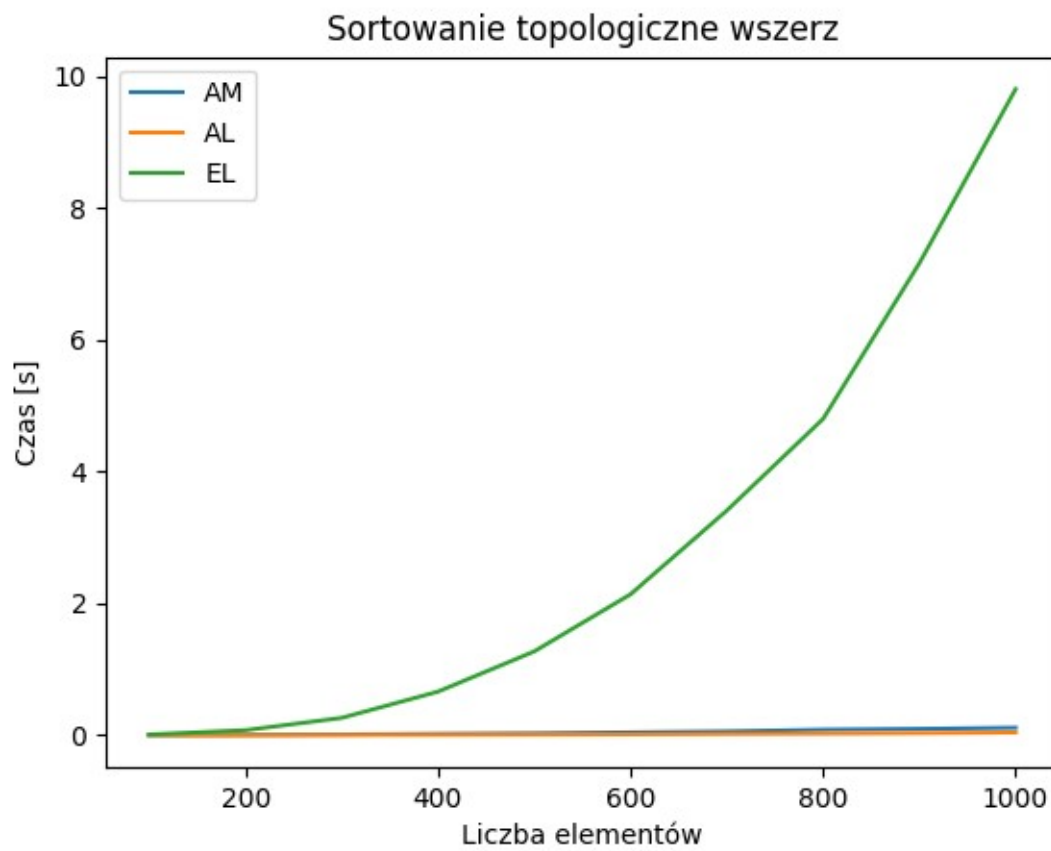


Figura 5: Sortowanie topologiczne BFS - wszystkie algorytmy

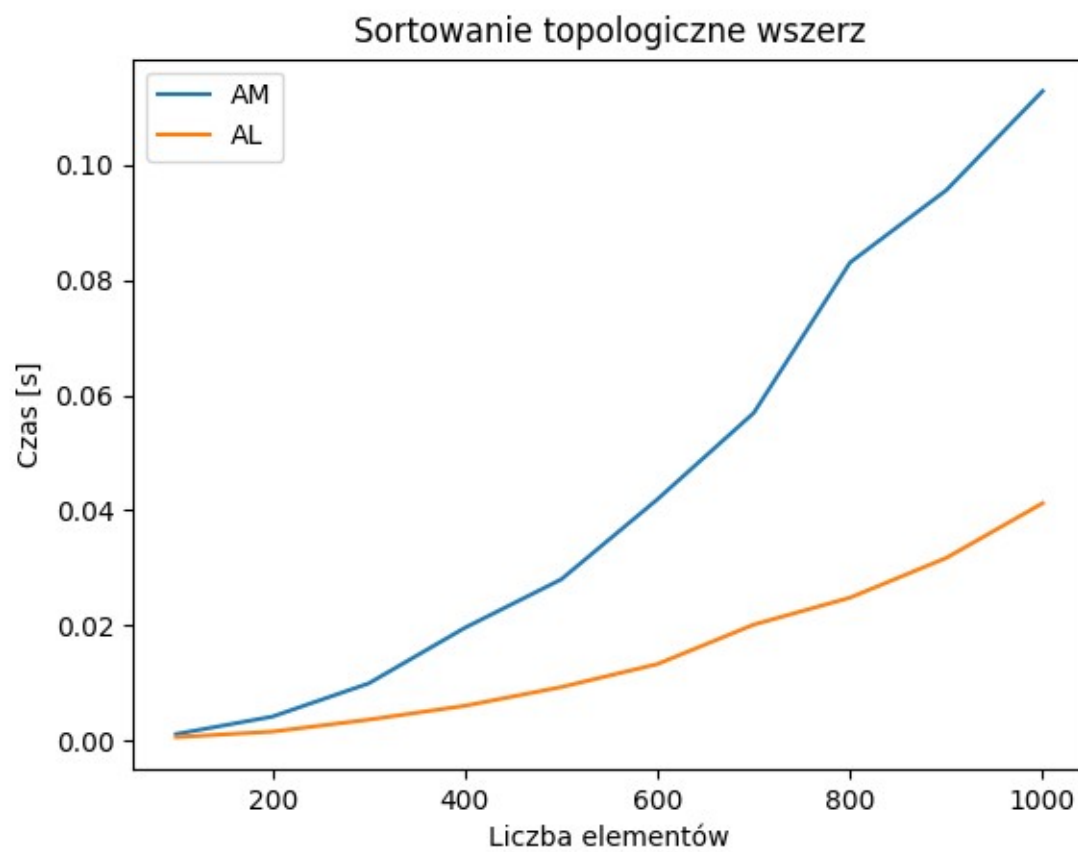


Figura 6: Sortowanie topologiczne BFS - bez EL

Sortowanie topologiczne DFS

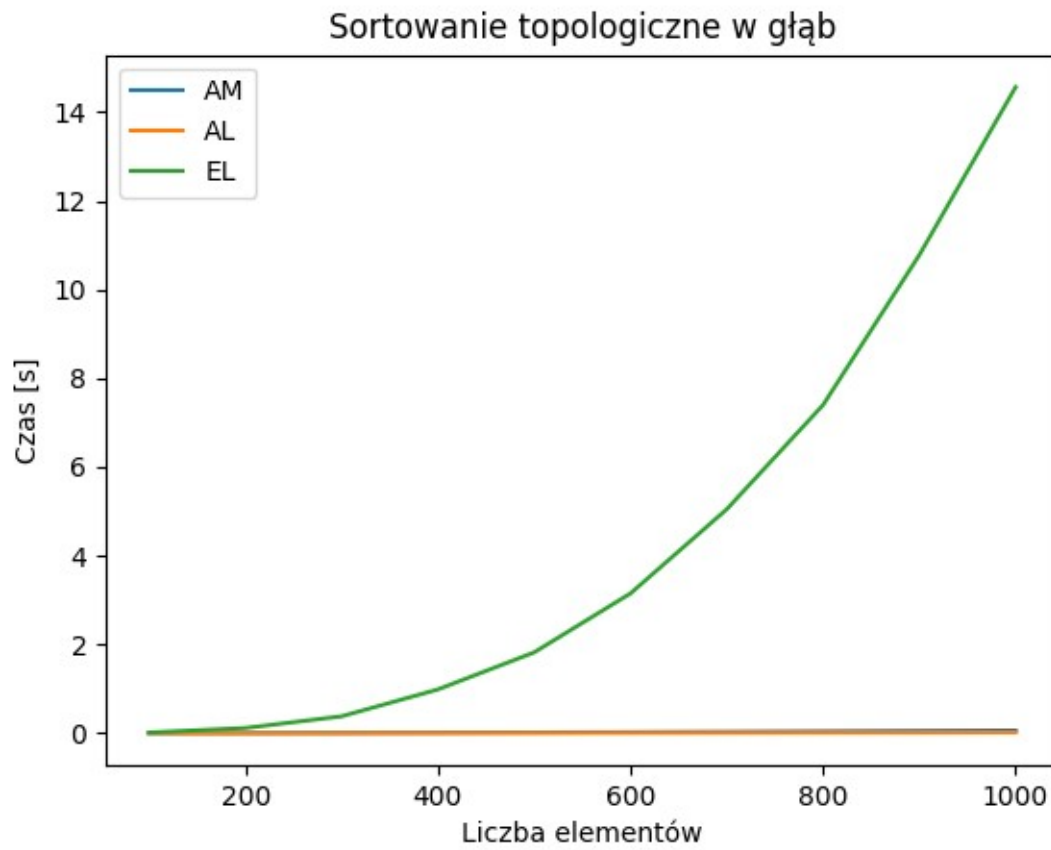


Figura 7: Sortowanie topologiczne DFS - wszystkie algorytmy

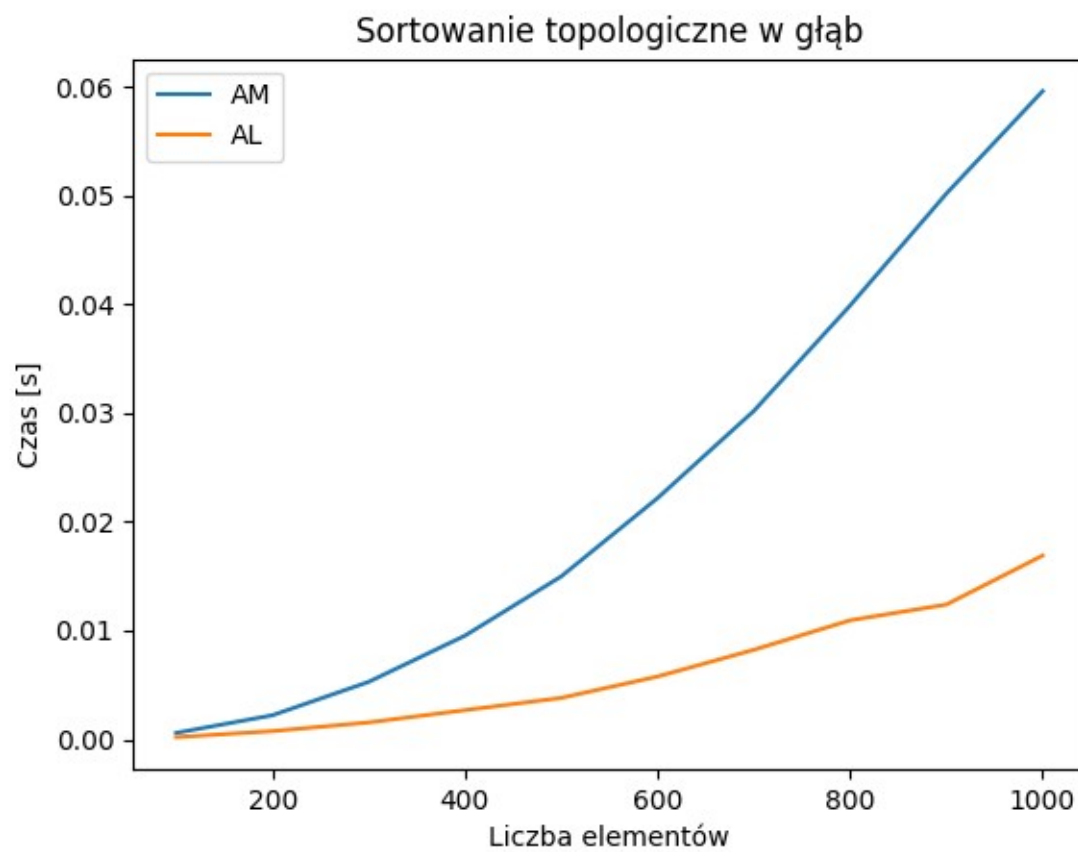


Figura 8: Sortowanie topologiczne DFS - bez EL