



Politechnika Poznańska
Wydział Informatyki i Telekomunikacji

Algorytmy i Struktury Danych, Informatyka (semestr 2)
Sprawozdanie #3 — Algorytmy grafowe

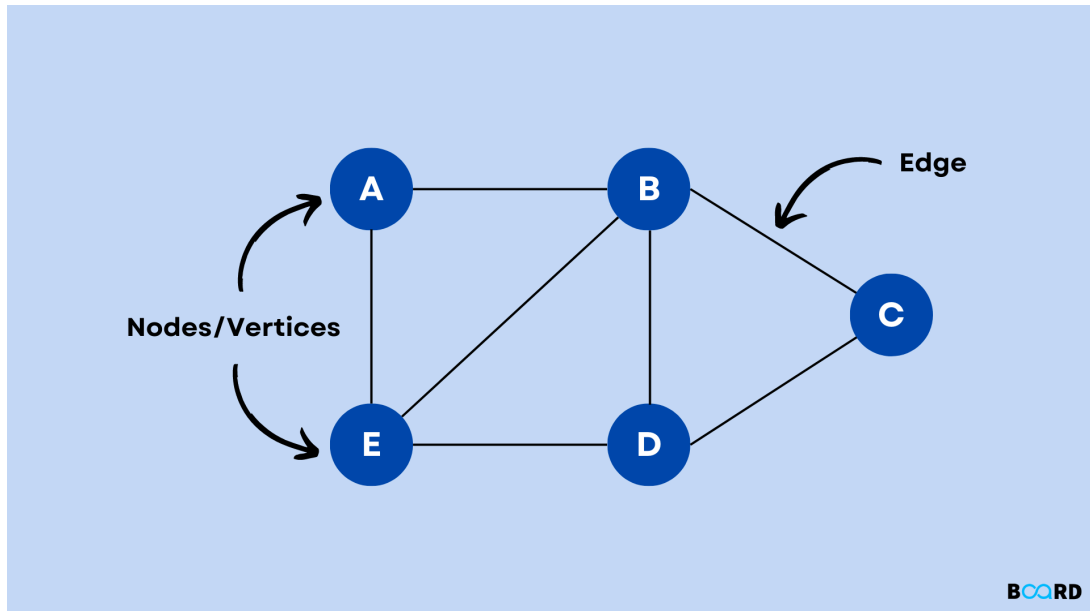


wykonał
Konrad Ceglarski
[156912]

1. Wprowadzenie

Graf to struktury danych, które reprezentują związki między obiektami.

Grafy zbudowane są z węzłów, zwanych też wierzchołkami (ang. nodes/vertices) oraz krawędzi, które łączą te wierzchołki.

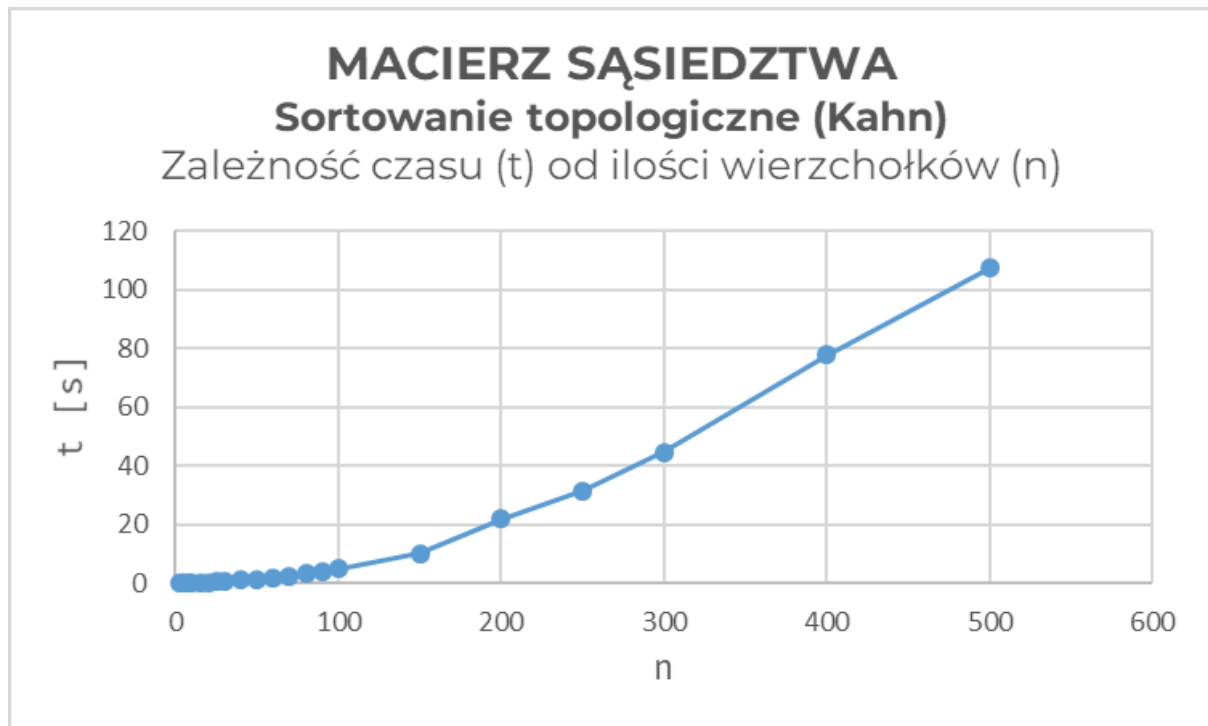


Grafy można reprezentować na różne sposoby zależnie od potrzeb i zastosowań. Najpopularniejsze reprezentacje grafów to:

- Macierz sąsiedztwa
- Lista następników
- Lista poprzedników
- Lista krawędzi
- Macierz incydencji

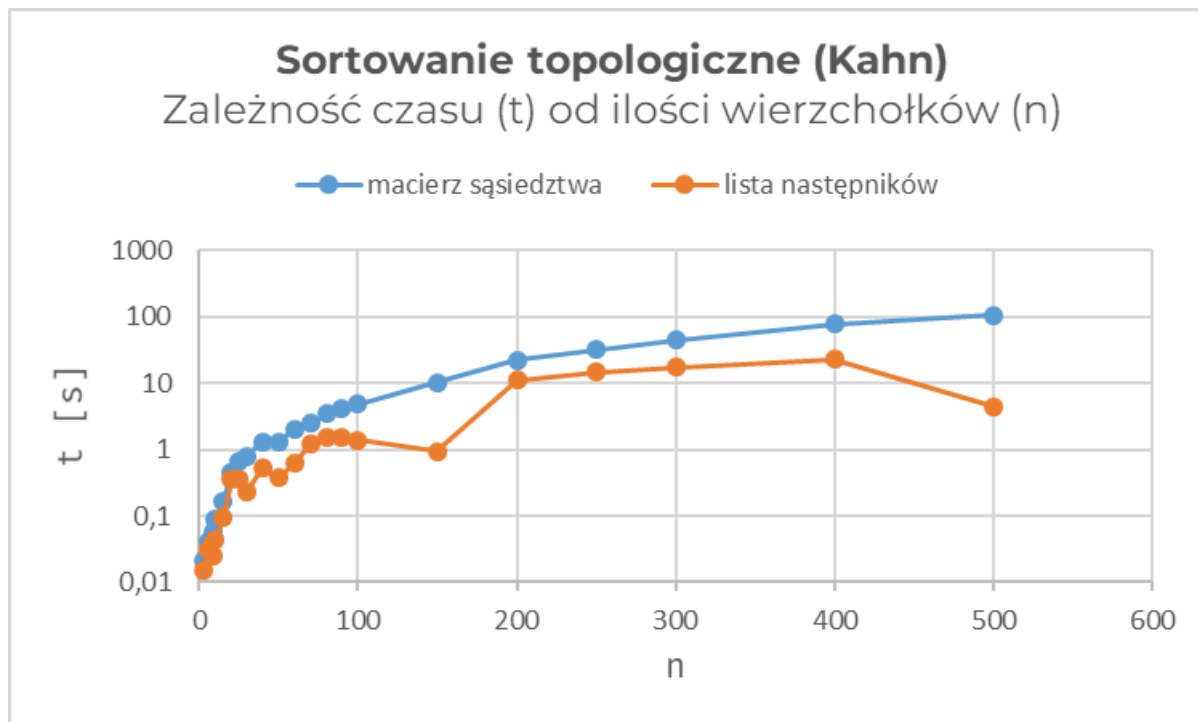
** Na potrzeby tego laboratorium skupię się na dwóch pierwszych, czyli macierzy sąsiedztwa oraz liście następników.*

2. Zależność czasu (t) od liczby węzłów (n) sortowania topologicznego dla obu reprezentacji



* Nieoczekiwane skoki najprawdopodobniej wynikają z losowych połączeń w grafach, które są niekorzystne dla danych reprezentacji.

3. Porównanie sortowania topologicznego dla obu reprezentacji w skali logarytmicznej



4. Zalety i wady obu reprezentacji grafu z punktu widzenia ich zastosowania w zaimplementowanych algorytmach.

Macierz sąsiedztwa	
Wady [-]	Zalety [+]
<ul style="list-style-type: none"> • Duże zapotrzebowanie na pamięć (marnotrawstwo pamięci w przypadku rzadkich grafów - wypełnianie macierzy zerami) • Sortowanie topologiczne wg. algorytmu Kahna jest wolniejsze niż w przypadku listy następników 	<ul style="list-style-type: none"> • Stały rozmiar • Bezpośredni dostęp (po indeksach) • Stabilny (gęstość i zróżnicowanie połączeń w grafie nie wpływa tak bardzo na czas sortowania)

Lista następników	
Wady [-]	Zalety [+]
<ul style="list-style-type: none"> • Mniej stabilny • Mniej bezpośredni dostęp (wymaga przeszukiwania) 	<ul style="list-style-type: none"> • Oszczędność pamięci • Efektywne przeszukiwanie następców • Sortowanie topologiczne wg. algorytmu Kahna jest szybsze niż w przypadku macierzy sąsiedztwa

Podsumowanie:

Obie reprezentacje mają swoje klasyczne dla świata algorytmów dylematy - pamięć, dokładność czy szybkość.

Z moich obserwacji wynika, że w używanych przeze mnie implementacjach algorytmu Kahna, lista następników jest zdecydowanie wydajniejsza (co widać na porównaniu w skali logarytmicznej).

Mimo to wykazana na moich implementacjach stabilność macierzy sąsiedztwa również zasługuje na wyróżnienie (czasy sortowania nie są tak wrażliwe na różnorodność grafów).

Język implementacji:

Python 🐍 (Python 3.10.10)

Platforma realizacji testów:

Windows 10 x64 (Microsoft Windows [Version 10.0.19045.4170])

Repozytorium:

https://github.com/xKond3i/put_aisd

Źródła:

[1]: cs.put.poznan.pl/mszachniuk/site/teaching/algorytmy-i-struktury-danych

[2]: www.ekursy.put.poznan.pl

[3]: www.eduinf.waw.pl/inf/alg

[4]: pl.wikipedia.org

[5]: chat.openai.com (implementacja algorytmu Kahna - z uwagi na niemożność implementacji osobiście)