

# Raport

Andrzej Swatowski (as386085)

## ZASTOSOWANE OPTYMALIZACJE

---

- **sortowanie list sąsiedztwa:**

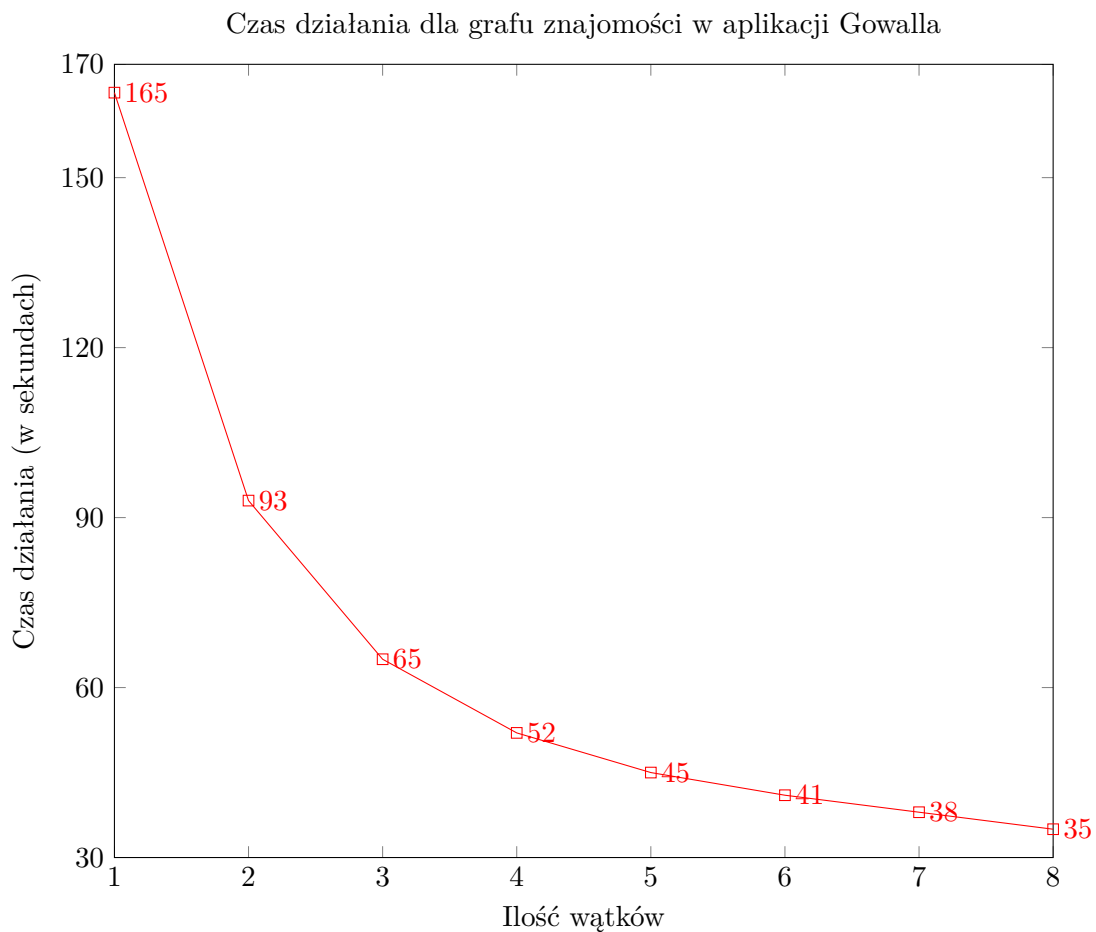
Po wczytaniu grafu z pliku program sortuje listy sąsiedztwa dla każdego z wierzchołków, malejąco względem wag krawędzi prowadzących do kolejnych sąsiadów. Nie jest to najbardziej optymalne rozwiązanie (w pracy opisującej algorytm przedstawione było szybsze - sortowanie tylko kilku pierwszych, największych sąsiadów), ale wciąż wystarczająco szybkie (a przy okazji łatwiejsze w implementacji). Dzięki temu każdy wierzchołek szuka sąsiada do adorowania najpierw spośród tych o najcięższych krawędziach (czyli tych najbardziej opłacalnych).

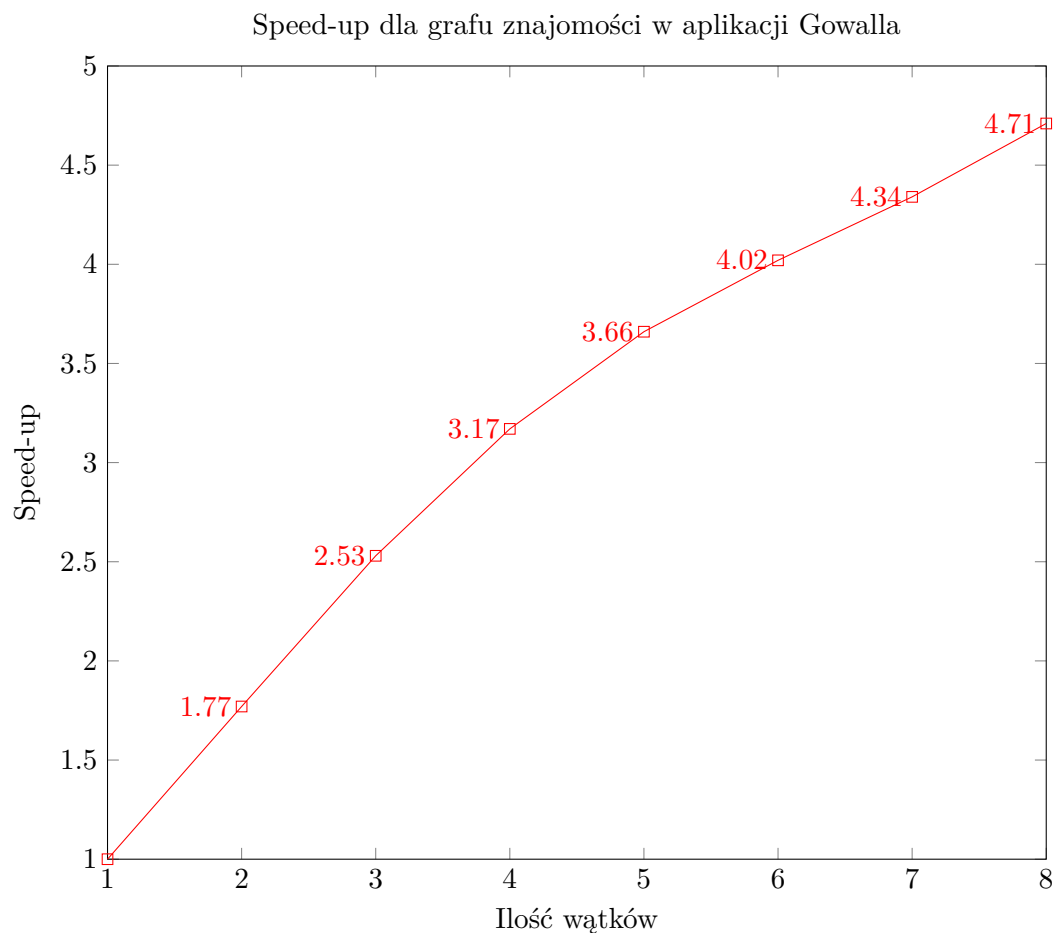
- **ustalenie kolejności przeglądania wierzchołków w grafie:**

Po wczytaniu grafu, przed wykonaniem algorytmu program sortuje wszystkie wierzchołki malejąco względem wag ich najcięższych krawędzi. Dzięki temu algorytm najpierw poszukuje sąsiadów do adorowania dla wierzchołków o najcięższych możliwych krawędziach, co skutkuje szybszym działaniem (propozycje „ciężkich” wierzchołków rzadziej są anulowane)

## PRZYSPIESZENIE

---





Wybrałem do testów graf loc-Gowalla ([loc-Gowalla na stronie SNAP](#)). Posiada on 196591 wierzchołków oraz 950327 krawędzi. Liczyłem czasy działania programu w sekundach za pomocą UNIXowego polecenia *time*, odejmując od ich średnich średni czas wczytywania danych.

Podczas liczenia dla  $\text{bmethod} = \{0 \dots 20\}$  przyspieszenie między 1 a 8 wątkami było ponad 4-krotne (z 2 minut 45 sekund do 35 sekund).

ilość wątków	czas działania	speed-up
1	2 minuty 45 sekund	1
2	1 minuta 33 sekund	1.77
3	1 minuta 5 sekund	2.53
4	52 sekund	3.17
5	45 sekund	3.66
6	41 sekund	4.02
7	38 sekund	4.34
8	35 sekund	4.71