Opracowanie algorytmów z powracaniem

Szymon Kaczmarek, nr indeksu 148056

1 Wprowadzenie

Niniejsze opracowanie ma na celu przedstawienie oraz przetestowanie dwóch algorytmów wykorzystujących powracanie; znajdowania cylku Eulera i cyklu Hamiltona w grafie nieskierowanym. Wszystkie programy zostały napisane w Pythonie a do reprezentowania grafu została wykorzystana lista sąsiedztwa (repozytorium GitHub).

2 Algorytmy

2.1 Cykl Eulera

Cykl Eulera w grafie to cykl, który zawiera wszystkie krawędzie grafu. Aby graf zawierał cykl Eulera musi być spełniony warunek taki, że wszystkie wierzchołki grafu mają parzysty stopień. Do znajdowania tego cyklu został wybrany algorytm Fleury'ego, który polega na przechodzeniu do kolejnych wierzchołków grafu i usuwaniu użytej do tego krawędzi. Algorytm się wykonuje aż do usunięcia wszystkich krawędzi. Złożoność czasowa tego algorytmu to $O(E)$, gdzie E oznacza liczbę krawędzi w grafie. Jest tak dlatego, że algorytm z każdym „krokiem" usuwa jedną krawędź, a wykonuje się on aż do usunięcia wszystkich krawędzi. Znajdowanie sąsiadów wykonuje się z złożonością $O(1)$, a usuwanie krawędzi z złożonością $O(E)$. Zaimplementowany algorytm zawiera również sprawdzenie czy jest spełniony warunek istnienia cyklu. Sprawdzenie to wykonuje się w czasie $O(V)$, gdzie V oznacza liczbę wierzchołków. Ogólnie, zaimplementowany algorytm działa w czasie $O(V + E)$.

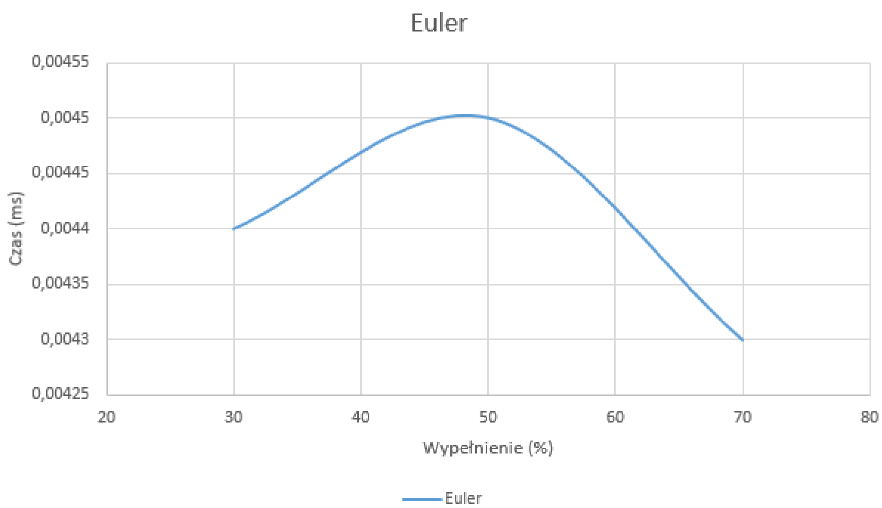
2.2 Cykl Hamiltona

Cykl Hamiltona to cykl, który zawiera wszystkie krawędzie grafu. Do wyszukania tego cyklo został zaimplementowany algorytm Robertsa-Floresa. Polega on na generowaniu kolejnych ścieżek i sprawdzeniu czy jest ona poszukiwanym cyklem. Algorytm ten posiada pesymistyczną złożoność $O(V!)$ ograniczoną przez ilość krawędzi, która reguluje ile ścieżek jest możliwych do uzyskania. Ciężko jest określić złożoność średnią z powodu, że problem ten jest NP-zupełny. Do tego wraz ze wzrostem krawędzi w grafie zachodzą ciekawe zjawiska odnośnie czasu wykonywania algorytmu. Optymistyczna złożoność algorytmu to $O(V)$ zakładając, że pierwsza znaleziona ścieżka jest poszukiwanym cyklem.

3 Czasy wykonywania

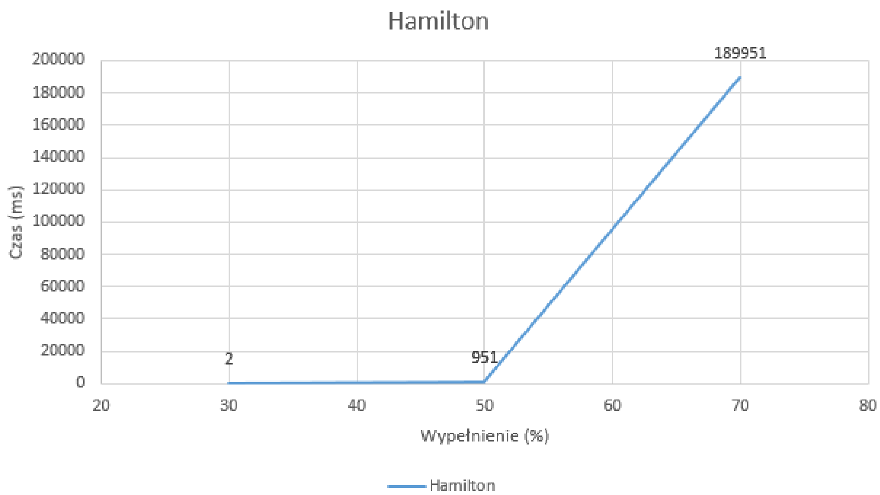
Grafy poddawane testom zawierały stałą liczbę wierzchołków i zwiększającą się liczbę krawędzi.

3.1 Algorytm poszukiwania cyklu Eulera dla grafu go nie posiadającego



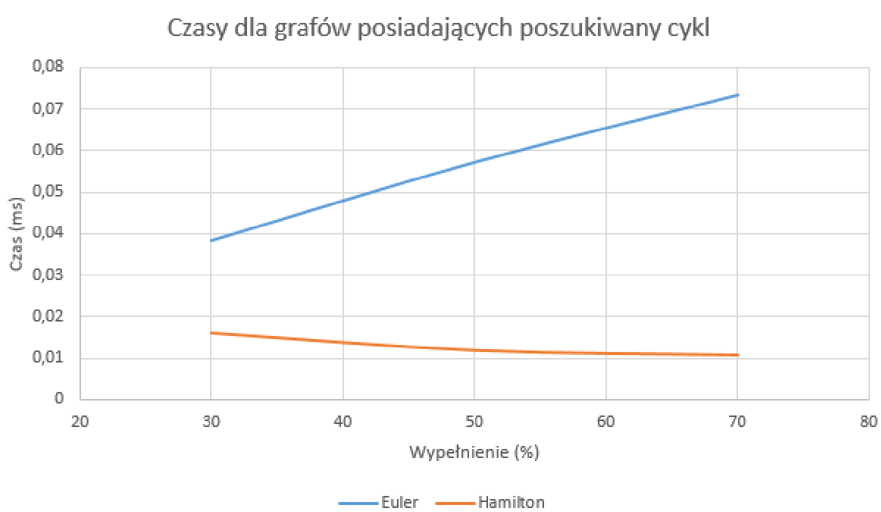
Z powodu wykorzystania sprawdzenia warunku koniecznego algorytm wykonuje się z czasem praktycznie stałym (różnica wartości to maksymalnie $0,0002ms$). Z powodu stałej liczby wierzchołków sprawdzenie tego warunku posiada względną złożoność $O(1)$.

3.2 Algorytm poszukiwania cyklu Hamiltona dla grafu go nie posiadającego



W przypadku nie występowania cyklu Hamiltona algorytm musi przeanalizować wszystkie możliwe ścieżki. Wszyskich ścieżek jest $V!$, jednakże, taką liczbę uzyskamy tylko przy wypełnieniu grafu w 100%. Wraz ze wzrostem ilości krawędzi znacząco wzrasta ilość ścieżek.

3.3 Algorytmy poszukiwania cyklów dla grafów je posiadających



Z wykresu można odczytać, że algorytm Fleury'ego posiada złożoność liniową.

Ciekawa rzecz dzieje się w przypadku cyklu Hamiltona. Możemy zauważyć spadek czasu wraz ze wzrostem ilości krawędzi. Spowodowane jest to tym, że przy większej ilości krawędzi pojawia się więcej możliwych cykli Hamiltona dlatego, szansa, że wybrana dotychczasowo ścieżka jest poprawna zwiększa się. Przy grafach rzadkich szansa ta jest mniejsza co powoduje, że algorytm musi więcej razy się cofnąć czyli zabiera więcej czasu. Jednakże, czasy te będą wzrastały do penego momentu a następnie znowu spadały ponieważ, przy grafach, które zawierają jeszcze mniejszą ilość krawędzi, algorytm posiada mniej możliwości podczas wyboru kolejnego wierzchołka. To sprawia, że możliwość pomyłki jest mniejsza, tak samo jak czas wykonywania.

4 Zakończenie

Zostały przedstawione, omówione i przetestowane algorytmy Fleury'ego oraz Robertsa-Floresa.

Spis treści