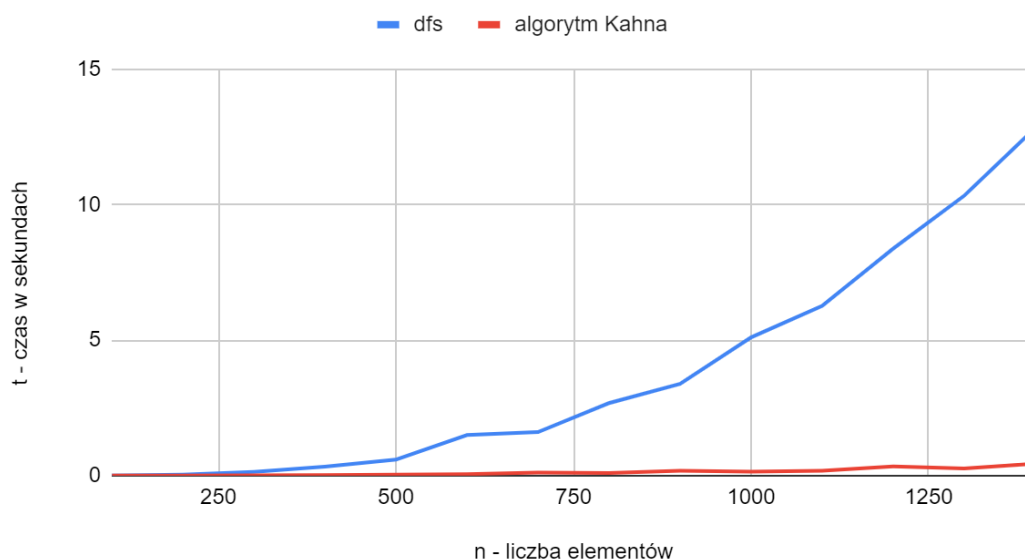


Algorytmy i struktury danych - Grafy

Aleksandra Ostrowska, nr 156121

W tym sprawozdaniu porównywane są działania różnych sposobów na przedstawienie grafu w macierzach: macierzy sąsiedztwa i macierzy grafu.

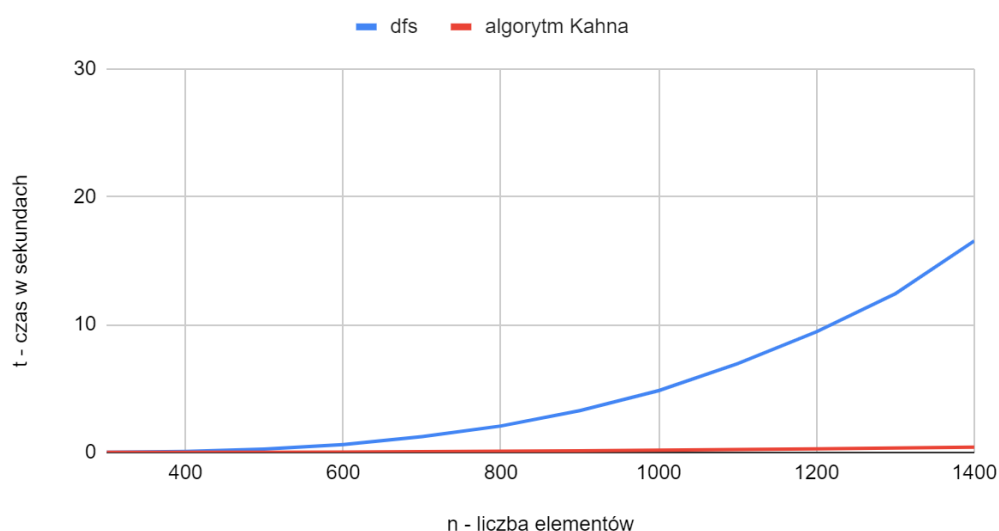
Sortowanie topologiczne macierzy sąsiedztwa $t=f(n)$



Rysunek 1: $t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w macierzy sąsiedztwa.

Macierz sąsiedztwa to macierz kwadratowa $[V \times V]$, gdzie krawędź między wierzchołkami jest przedstawiona za pomocą 1, a brak 0. Jest to reprezentacja maszynowa prosta w implementacji.

Sortowanie topologiczne macierzy grafu $t=f(n)$



Rysunek 2: $t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w macierzy grafu.

Macierz grafu to macierz o wymiarach $[V \times V + 3]$, gdzie w $[V+1]$ kolumnie znajduje się pierwszy następnik, w kolumnie $[V+2]$ pierwszy poprzednik, a w $[V+3]$ pierwszy wierzchołek z brakiem incydencji. W samej macierzy $[V \times V]$ znajduje się sieć następników, poprzedników i braku incydencji. Elementy są ustawione w taki sposób, że wskazują na siebie wzajemnie, tak że przechodząc przez jeden wiersz poznajemy wszystkie jego następniki, poprzedniki i wierzchołki, z którymi nie ma incydencji.

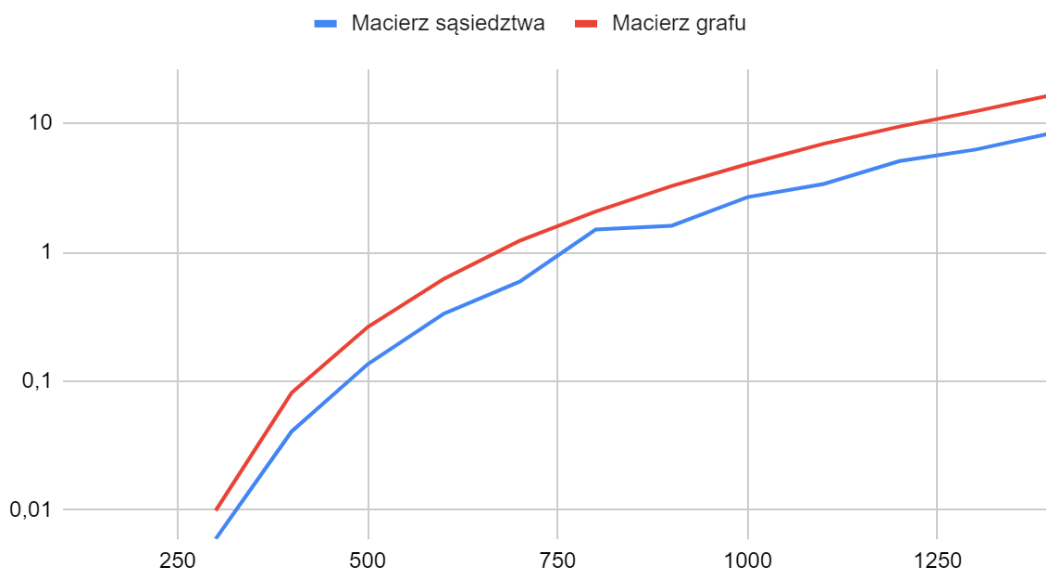
Ze względu na konstrukcję tej macierzy, implementacja jest bardziej skomplikowana za to prostymi działaniami otrzymujemy znacznie więcej informacji niż w macierzy sąsiedztwa.

Powyżej jest przedstawione sortowanie topologiczne dwoma metodami, algorytmami Kahna i DFS. Algorytm Kahna jest zauważalnie szybszy niż algorytm DFS.

Algorytm Kahna polega na tworzeniu listy wierzchołków, które nie mają żadnych krawędzi wchodzących do nich. Następnie usuwa się wierzchołki "zerowego stopnia do" oraz ich krawędzie, a powstałe w ten sposób nowe wierzchołki o zerowym stopniu dodaje się do listy. Proces ten powtarza się, aż wszystkie wierzchołki zostaną usunięte z grafu.

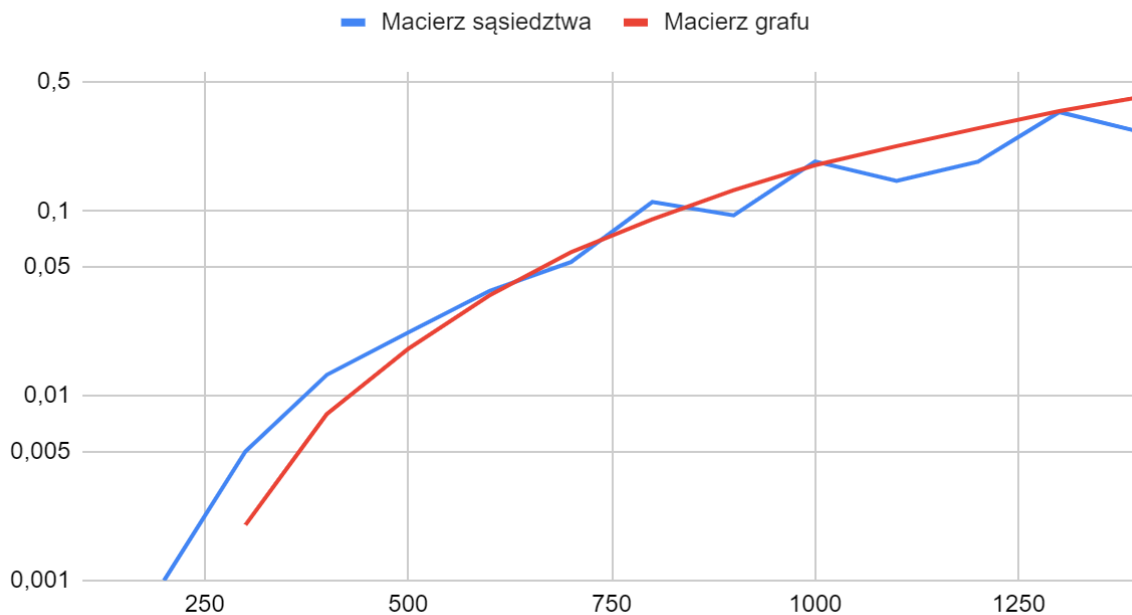
Natomiast algorytm DFS polega na przeszukiwaniu grafu w głąb, zaczynając od jednego wierzchołka i odwiedza kolejne wierzchołki w głąb. Algorytm zaznacza odwiedzone wierzchołki i powraca do poprzedniego wierzchołka, gdy wszystkie wierzchołki w tej ścieżce zostały oznaczone jako odwiedzone.

Działanie algorytmu DFS na macierzach $t=f(n)$



Rysunek 3: Zależność czasu obliczeń t od liczby n wierzchołków w grafie dla algorytmu Depth Search First.

Działanie algorytmu Kahna na macierzach $t=f(n)$



Rysunek 4: Zależność czasu obliczeń t od liczby n wierzchołków w grafie dla algorytmu Kahna.

Porównując te dwie reprezentacje maszynowe, ich złożoność obliczeniowa możemy wyciągnąć takie wnioski. Wybór macierzy zależy od tego, jakich informacji będziemy potrzebować. Jeśli naszym celem jest przejście wszystkich krawędzi, bardziej skuteczna jest macierz grafu, której zajmuje to średnio $O(V)$, gdzie w macierzy sąsiedztwa jest to $O(V^2)$. Jeśli istotne dla nas są listy następników i poprzedników powinniśmy także sięgnąć po macierz grafu. Niestety gdzie dodawanie krawędzi w macierzy sąsiedztwa jest szybkie tak w macierzy grafu zajmuje więcej czasu, ponieważ trzeba spełnić jej wiele warunków.

Ale porównując sortowanie topologiczne pomiędzy dwoma macierzami, czas wykonywania jest bardzo podobny. DFS jest nieznacznie wolniejszy dla macierzy grafu.

Algorytmy i struktury danych - Cykle Eulera i Hamiltona

Do znalezienia cyklu hamiltona i eulera w grafie został wykorzystany algorytm DFS (Deep First Search), w tym do poszukiwania cyklu hamiltona wykorzystano algorytm z powracaniem.

Dla cyklu Eulera klasa złożoności problemu decyzyjnego, jak i przeszukiwania należy do klasy P, jest to problem łatwy obliczeniowo. Złożoność algorytmów jest to złożoność wielomianowa.

Dla algorytmu szukającego cyklu Eulera, który przegląda sąsiadów wierzchołka i po kolei usuwa wspólne krawędzie to $O(|V|^2)$.

Dla cyklu Hamiltona, problem już nie jest tak prosty. Klasa złożoności problemu należy do klasy problemów silnie NP-trudnych dla problemu przeszukiwania, a wersja decyzyjna do klasy problemów silnie NP-zupełnych. Złożoność obliczeniowa tych algorytmów nie jest wielomianowa.

Ze względu na komplikacje w znalezieniu cyklu hamiltona złożoność obliczeniowa operacji może wynosić $O(|V|!)$.

Jak algorytmy radzą sobie z różnymi stopniami nasycenia?

Dla grafów o niskim nasyceniu, poszukując cyklu eulera, gdzie liczba krawędzi jest nieznaczna, algorytm może działać efektywnie, ponieważ większość wierzchołków nie będzie miała wiele sąsiadów do odwiedzenia. Z drugiej strony dla grafów gęstych, ze względu na dużą ilość krawędzi do odwiedzenia czas wyszukiwania cyklu eulera zostaje wydłużony.

W przypadku cyklu hamiltona, algorytm może napotkać wiele możliwych ścieżek i cykli Hamiltona. Często będzie musiał dokonać wielu prób i rekurencyjnych wywołań, zanim odnajdzie pełen cykl, co może prowadzić do większego czasu wykonania.

Dla cyklu hamiltona grafy o małym nasyceniu krawędziami stanowią najgorsze przypadki. Wtedy czas znajdowania cyklu hamiltona jest niestabilny i może zabierać czas powyżej minuty, co wynika z ewentualnych odchyśleń spowodowanych losowością wygenerowanych grafów. Ze wzrostem ilości krawędzi, spada czas wymagany do znalezienia cyklu.

Reprezentacje maszynowe grafu:

W przypadku wyszukiwania cyklu Hamiltona:

Lista następników może być bardziej efektywna dla grafów rzadkich, gdzie liczba krawędzi jest znacznie mniejsza niż liczba wierzchołków (lista następników zabiera mniej pamięci od macierzy sąsiedztwa). Przy użyciu listy następników możemy łatwo iterować po sąsiadach danego wierzchołka i sprawdzać istnienie krawędzi między nimi.

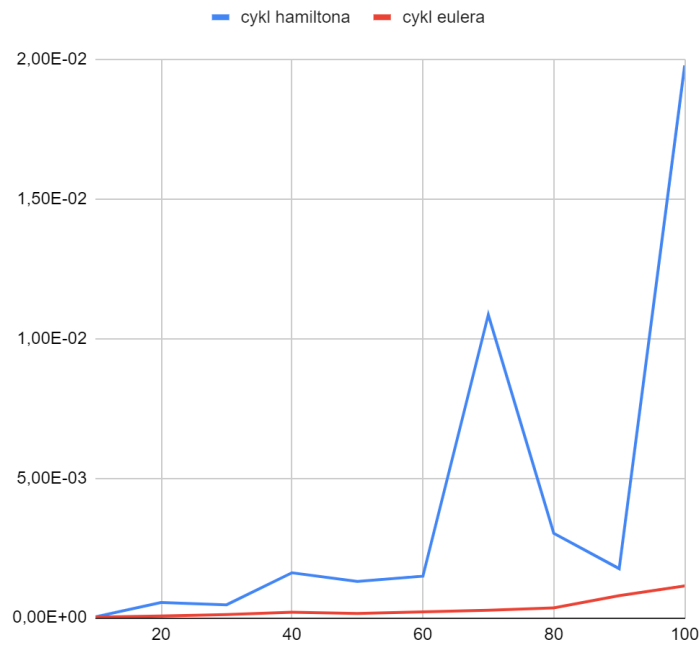
Macierz sąsiedztwa może być bardziej wydajna dla grafów gęstych, gdzie większość par wierzchołków jest połączona krawędzią. Dostęp do krawędzi między dwoma wierzchołkami w macierzy sąsiedztwa jest szybki i działa w czasie stałym $O(1)$.

W przypadku wyszukiwania cyklu Eulera:

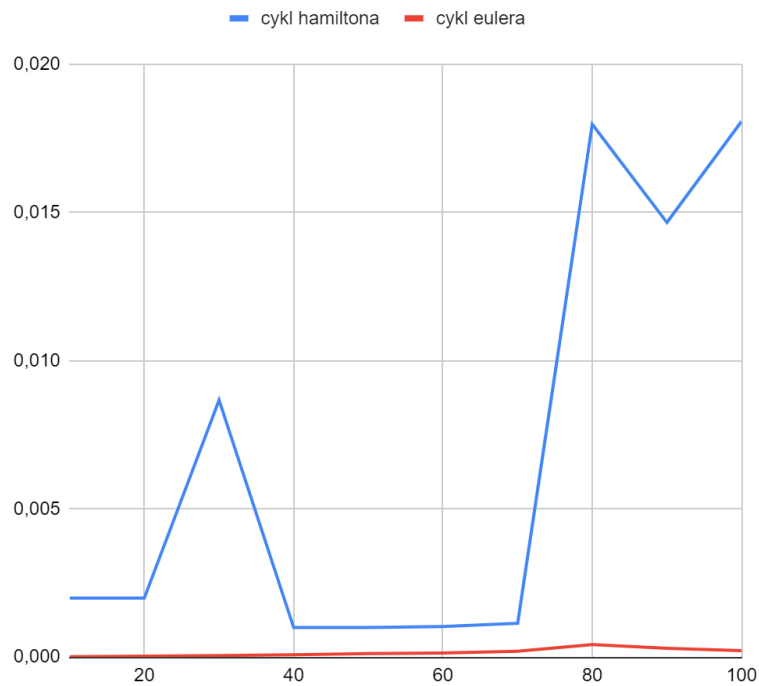
Macierz sąsiedztwa może być bardziej wygodna, ponieważ przechowuje informacje o istnieniu lub braku krawędzi między wierzchołkami. Możemy łatwo usunąć odwiedzone już krawędzie z grafu.

Lista następników może być również używana, ale może wymagać dodatkowych operacji, takich jak usuwanie węzłów z listy sąsiadów, np. dokonywać dodatkowych operacji wrzucania obecnie korzystanych wierzchołków na stos.

$t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie nieskierowanym przy stałej wartości nasycenia $s=50\%$

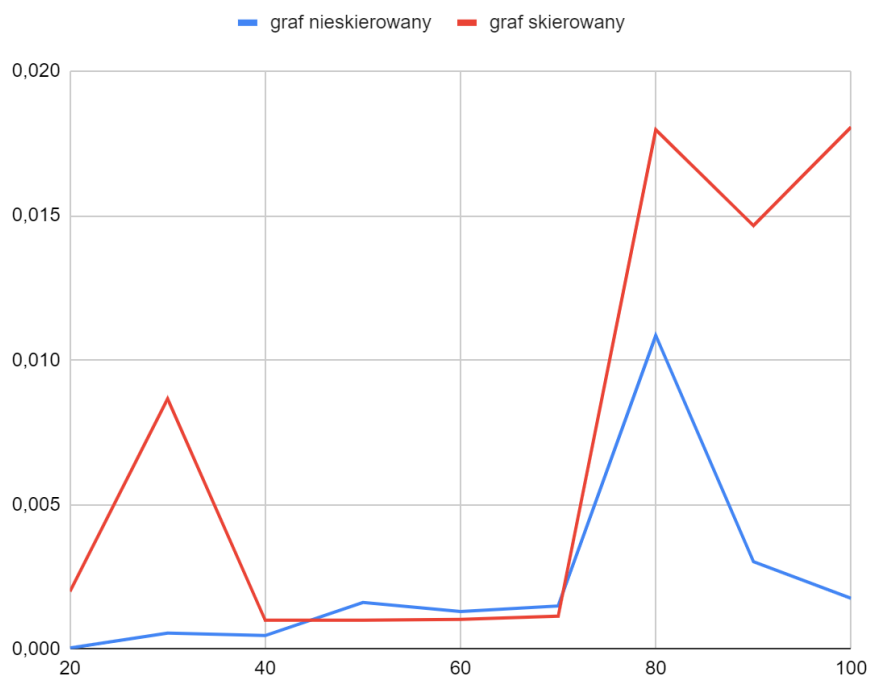


$t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie skierowanym przy stałej wartości nasycenia $s=50\%$

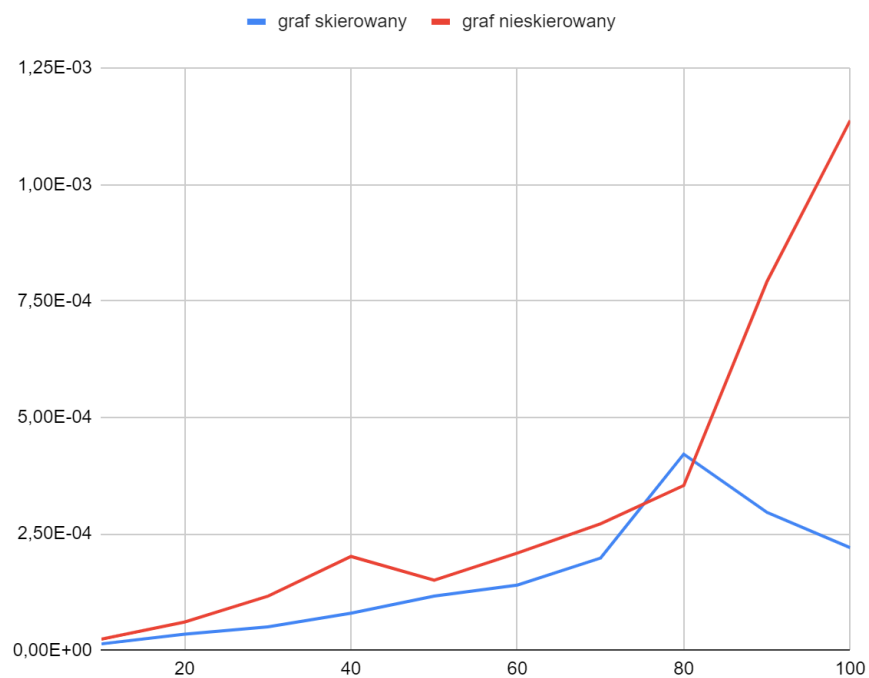


Rysunek 5 i 6: $t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie przy stałej wartości nasycenia $s=50\%$. Porównanie algorytmów.

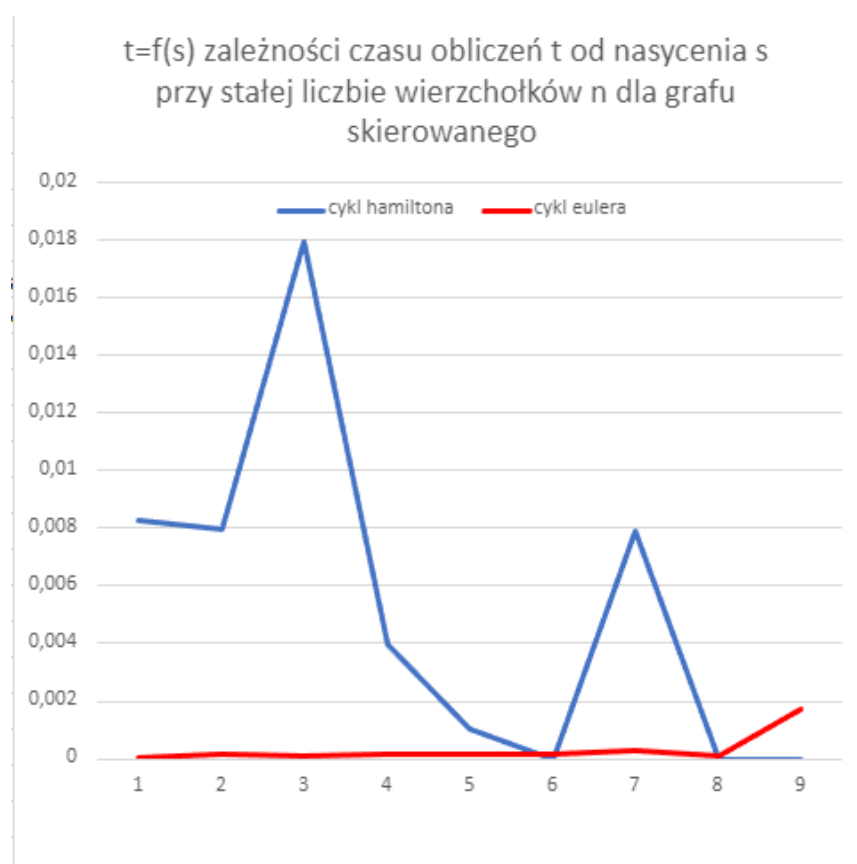
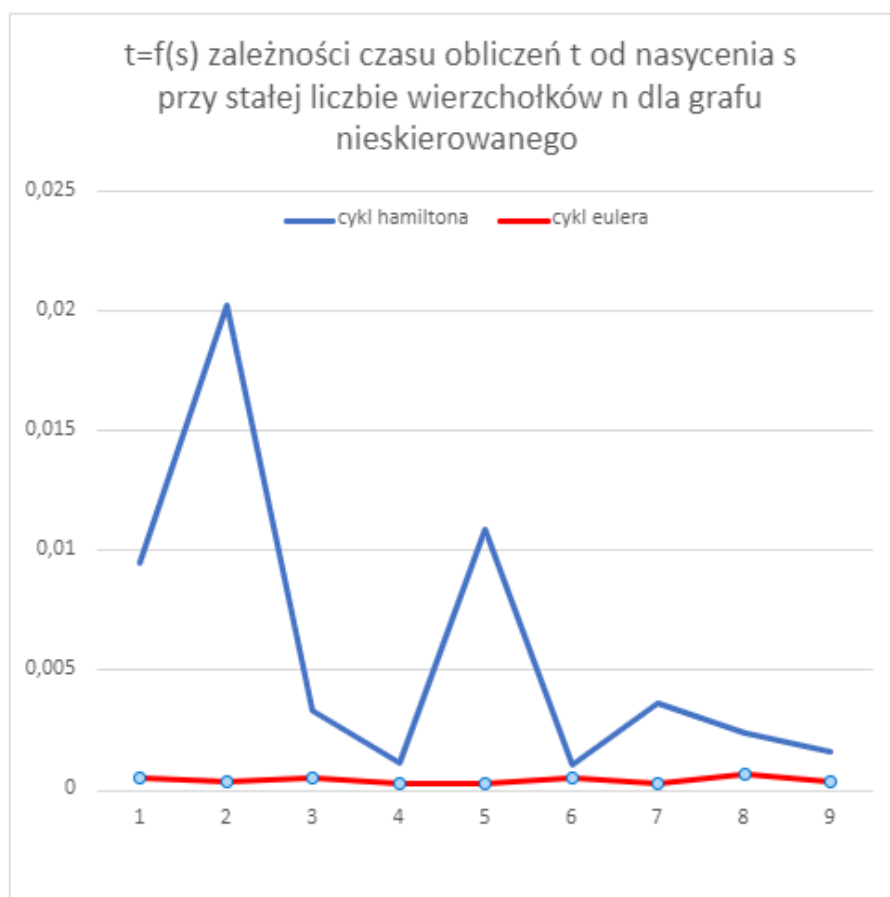
Cykl hamiltona, $t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie przy stałej wartości nasycenia $s=50\%$.



Cykl eulera, $t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie przy stałej wartości nasycenia $s=50\%$.

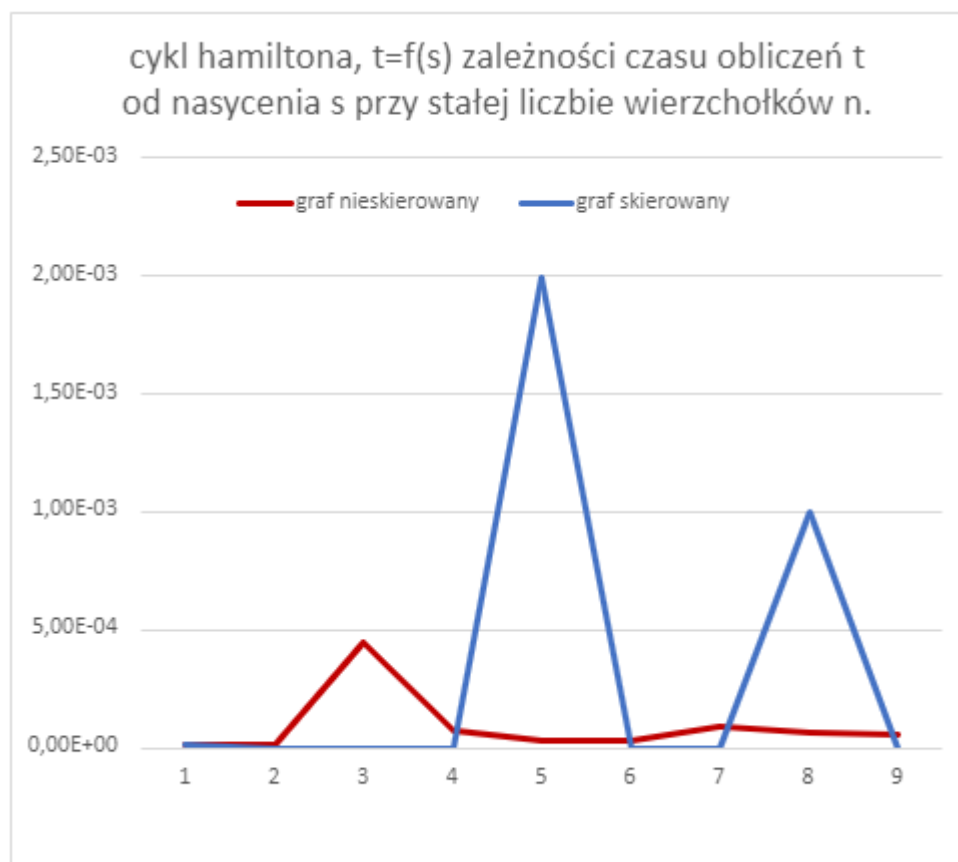
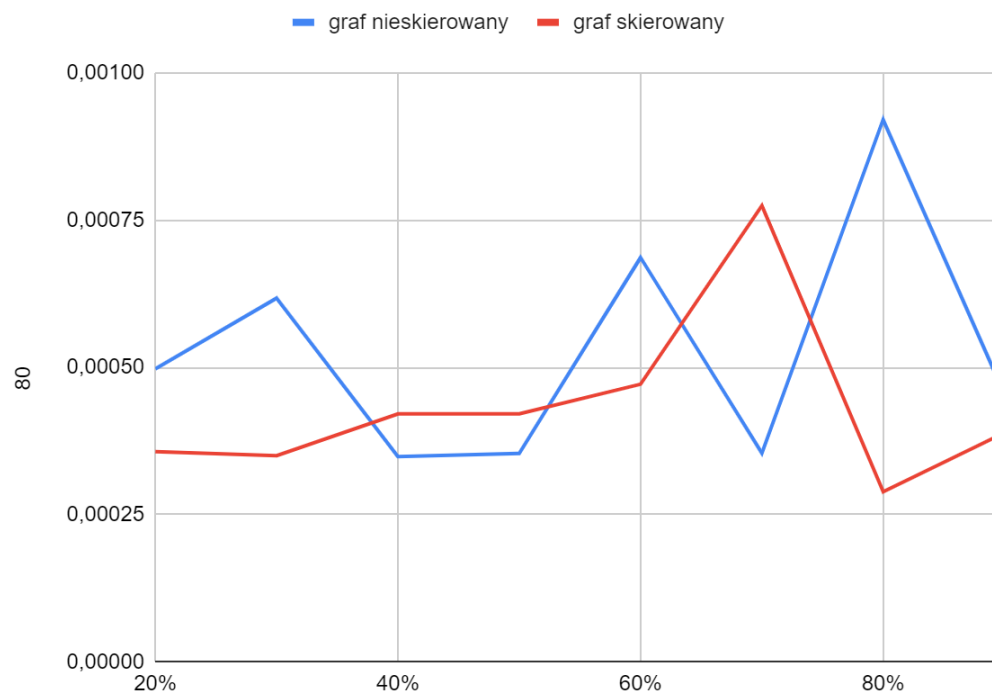


Rysunek 7 i 8: $t=f(n)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie przy stałej wartości nasycenia $s=50\%$. Porównanie reprezentacji grafu.



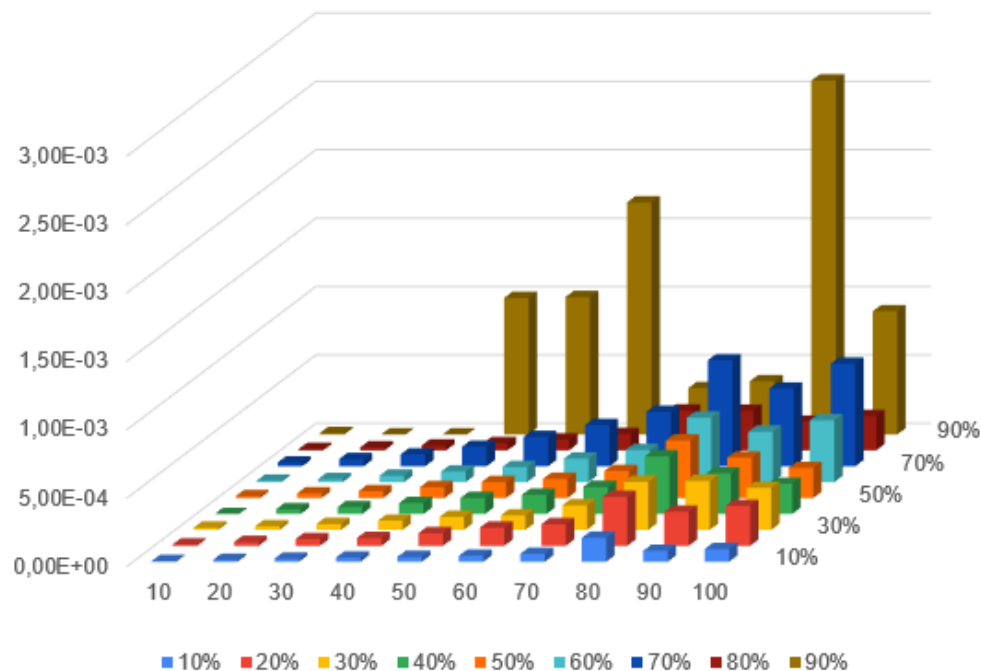
Rysunek 9 i 10: $t=f(s)$ zależności czasu obliczeń t od nasycenia s przy stałej liczbie wierzchołków n .
Porównanie algorytmów.

cykl eulera, $t=f(s)$ zależności czasu obliczeń t od nasycenia s przy stałej liczbie wierzchołków n .

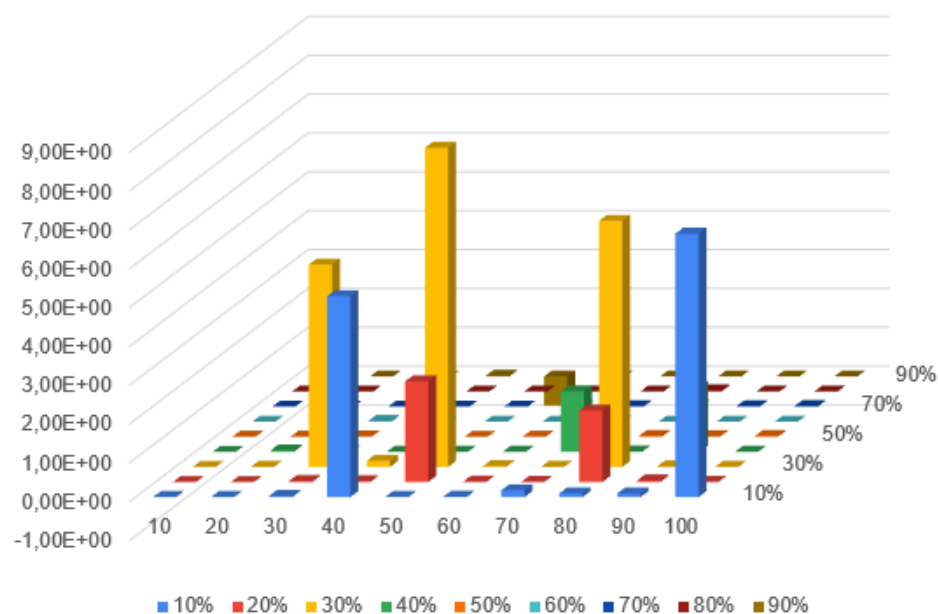


Rysunek 11 i 12: $t=f(s)$ zależności czasu obliczeń t od nasycenia s przy stałej liczbie wierzchołków n .
Porównanie reprezentacji grafu.

$t=f(n,s)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie i nasycenia s , dla poszukiwania cyklu eulera w grafie skierowanym.

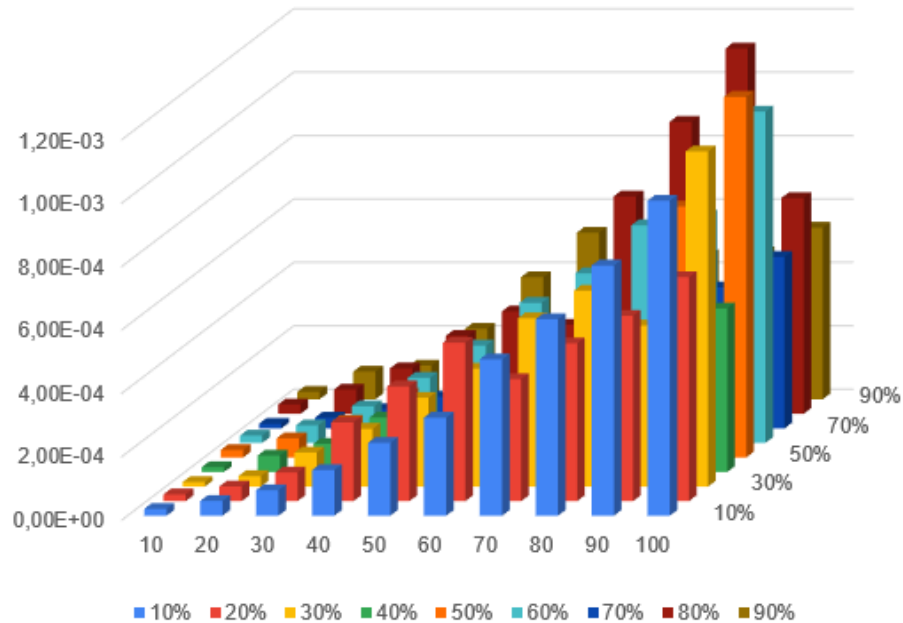


$t=f(n,s)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie i nasycenia s , dla poszukiwania cyklu hamiltona w grafie skierowanym.

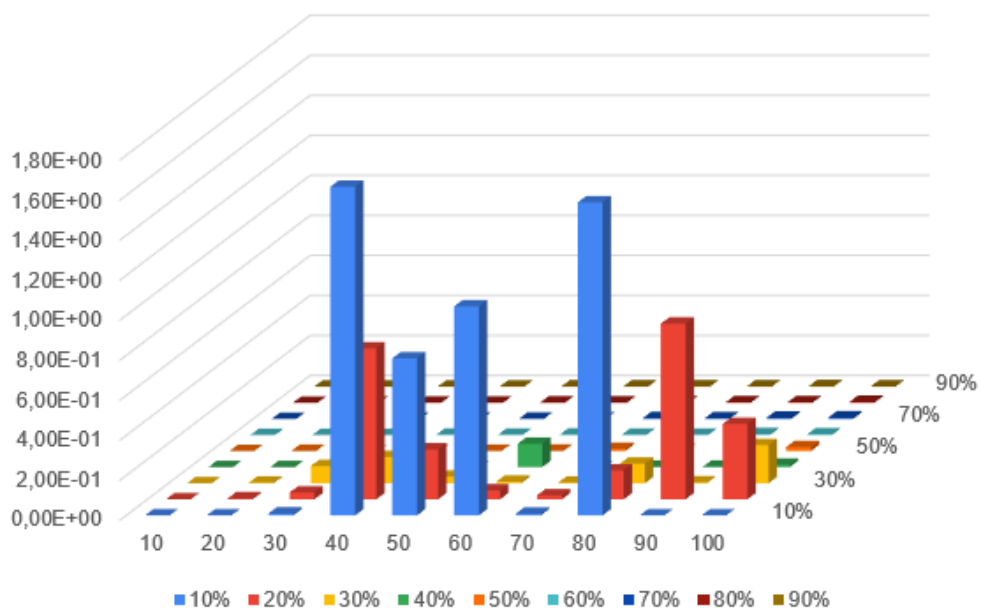


Rysunek 13 i 14: $t=f(n,s)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie i nasycenia s , dla poszukiwania cyklu w grafie skierowanym.

$t=f(n,s)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie i nasycenia s , dla poszukiwania cyklu eulera w grafie nieskierowanym.



$t=f(n,s)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie i nasycenia s , dla poszukiwania cyklu hamiltona w grafie nieskierowanym.



Rysunek 15 i 16: $t=f(n,s)$ zależności czasu obliczeń t od liczby n wierzchołków w grafie i nasycenia s , dla poszukiwania cyklu w grafie nieskierowanym