

## SPRAWOZDANIE 4

### Wprowadzenie:

Celem niniejszego sprawozdania jest zbadanie złożoności obliczeniowej, oraz omówienie zasad działania poszczególnych algorytmów szukania drogi eulera i ścieżki hamiltona dla grafów skierowanych i nieskierowanych w reprezentacjach grafu odpowiednio: listy następników, macierzy sąsiedztwa. Wszystkie algorytmy były sprawdzano za pomocą, wcześniej wczytanych wartości z pliku tekstowego zawierającego informacje o połączeniach między wierzchołkami, liczbie wierzchołków, oraz liczbę krawędzi.

```
def macierz_sasiedztwa_plik(file_path):
    with open(file_path, 'r') as file:
        num_vertices, num_edges = map(int, file.readline().split())
        graph = [[0] * num_vertices for _ in range(num_vertices)]
        for _ in range(num_edges):
            edge = file.readline().split()
            out, inn = map(int, edge)
            graph[out - 1][inn - 1] = 1
            graph[inn - 1][out - 1] = 1
        return graph

#GENERATOR MACIERZY SASIEDZTWA
def macierz_sasiedztwa_generator(liczba_wierzchołkow, liczba_krawedzi):
    #tworzenie pustej macierzy wypełnionej zerami
    matrix = []
    for i in range(liczba_wierzchołkow):
        row = []
        for j in range(liczba_wierzchołkow):
            row.append(0)
        matrix.append(row)

    # Iterujemy się przez wszystkie krawędzie, które chcemy dodać.
    for _ in range(liczba_krawedzi):
        while True: # Pętla, która będzie powtarzać losowanie, dopóki nie znajdziemy
            odpowiedniej krawędzi.
            wiersz = random.randint(0, liczba_wierzchołkow - 1)
            kolumna = random.randint(0, liczba_wierzchołkow - 1)
            # Sprawdzamy, czy wylosowana krawędź już istnieje (wartość różna od 0) oraz czy
            nie tworzy pętli.
            if wiersz != kolumna and matrix[wiersz][kolumna] == 0:
                # Jeśli warunki są spełnione, ustawiamy krawędź między wierzchołkami na 1.
                matrix[wiersz][kolumna] = 1
                matrix[kolumna][wiersz] = 1
                break # Przerywamy pętlę while i przechodzimy do kolejnej krawędzi.

    return matrix # Zwracamy wygenerowaną macierz sasiedztwa.
```

### PRZYKŁADOWO:

3 4  
1 2  
2 1  
4 2  
4 1

Pierwsza linia oznacza, że graf zawiera "3 wierzchołki i 4 krawędzie", każda kolejna oznacza krawędź "a b" z wierzchołka "a" do "b". Złożoność obliczeniowa została sprawdzona i sporządzona na wykresach za pomocą zewnętrznej biblioteki *timeit* służącej do wykonywania wykresów, oraz w excelu. W sprawozdaniu zostały

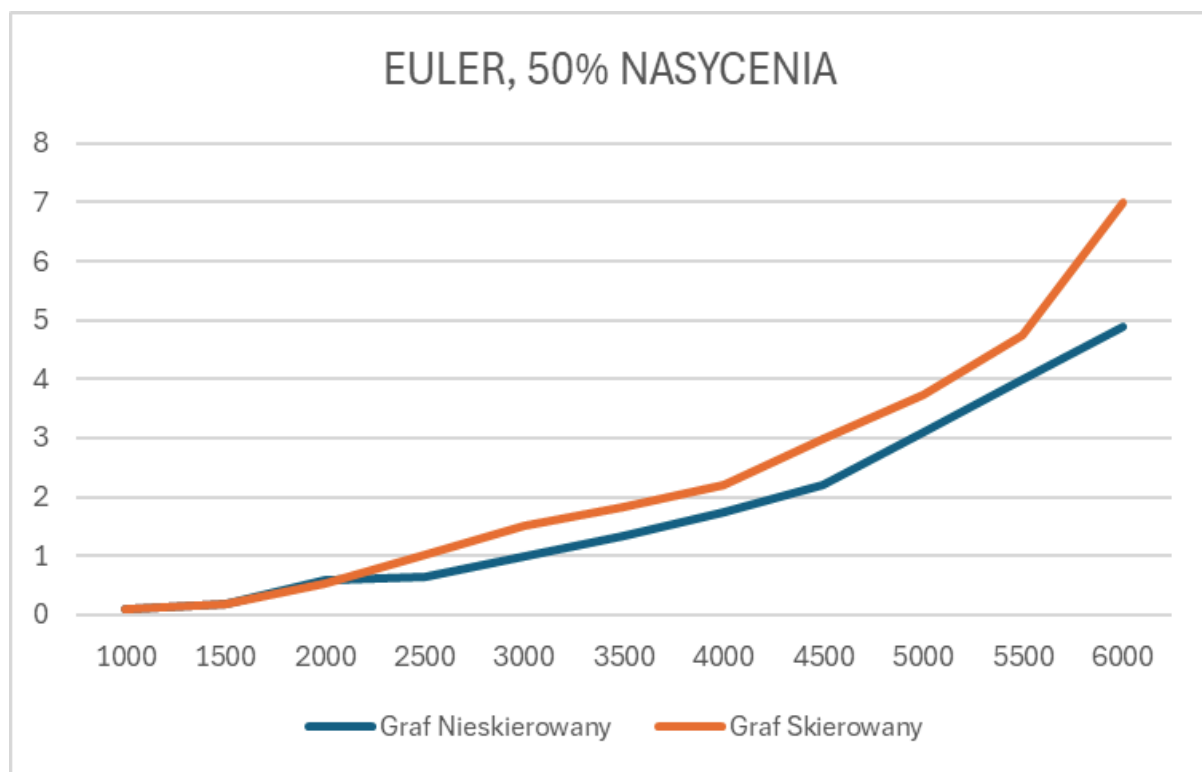
zawarte po dwa algorytmy, znajdowania cyklu hamiltona i eulera dla każdego rodzaju macierzy (macierzy sąsiedztwa listy następników). W pierwszej sekcji zostały zawarte wykresy zależności czasu od liczby elementów dla poszczególnych algorytmów. Oraz krótkie omówienie jak działają poszczególne funkcje.

## SEKCJA 1

### CYKL EULERA

Zależność czasu obliczeń  $t$  od liczby  $n$  wierzchołków w grafie przy stałej wartości nasycenia  $s = 50\%$ .

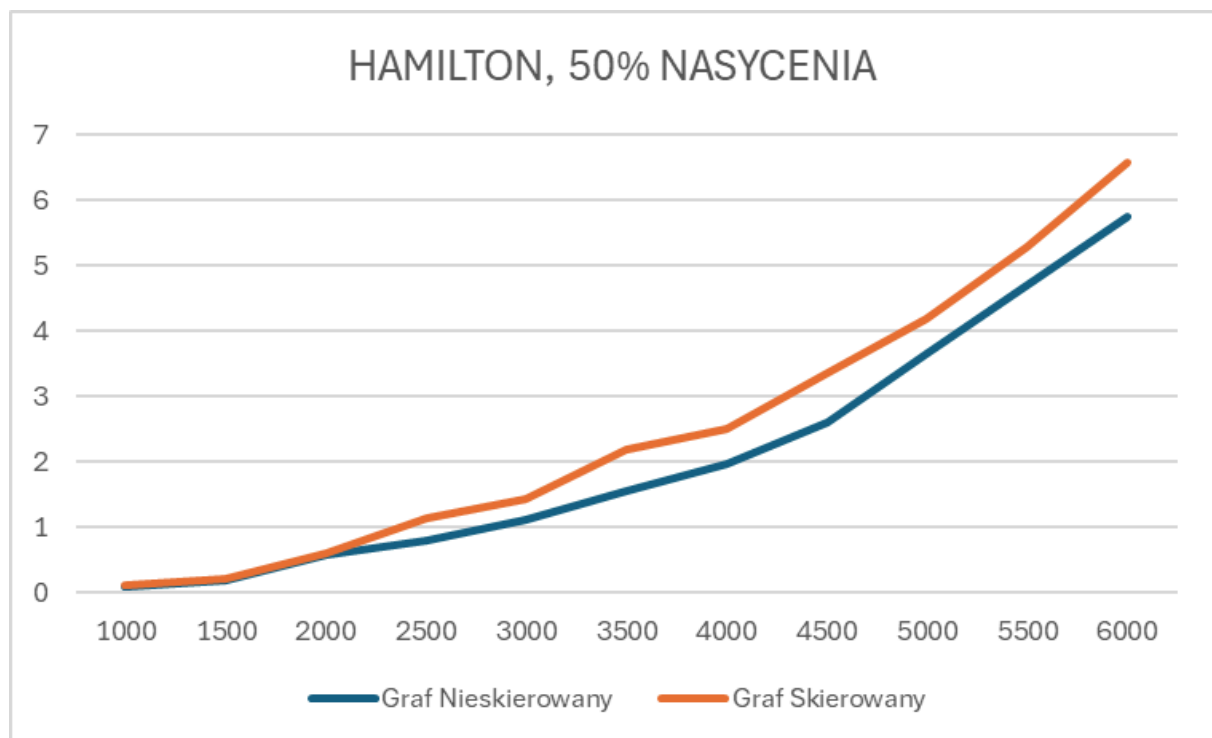
	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000
Graf Nieskierowany	0,0812	0,1793	0,57512	0,62951	0,979	1,34331	1,7512	2,2129	3,1	3,991	4,9012
Graf Skierowany	0,092771	0,18432	0,527276	1,00404	1,50448	1,81568	2,21512	2,98039	3,73973	4,74078	7,01105



## CYKL HAMILTONA

Zależność czasu obliczeń  $t$  od liczby  $n$  wierzchołków w grafie przy stałej wartości nasycenia  $s = 50\%$ .

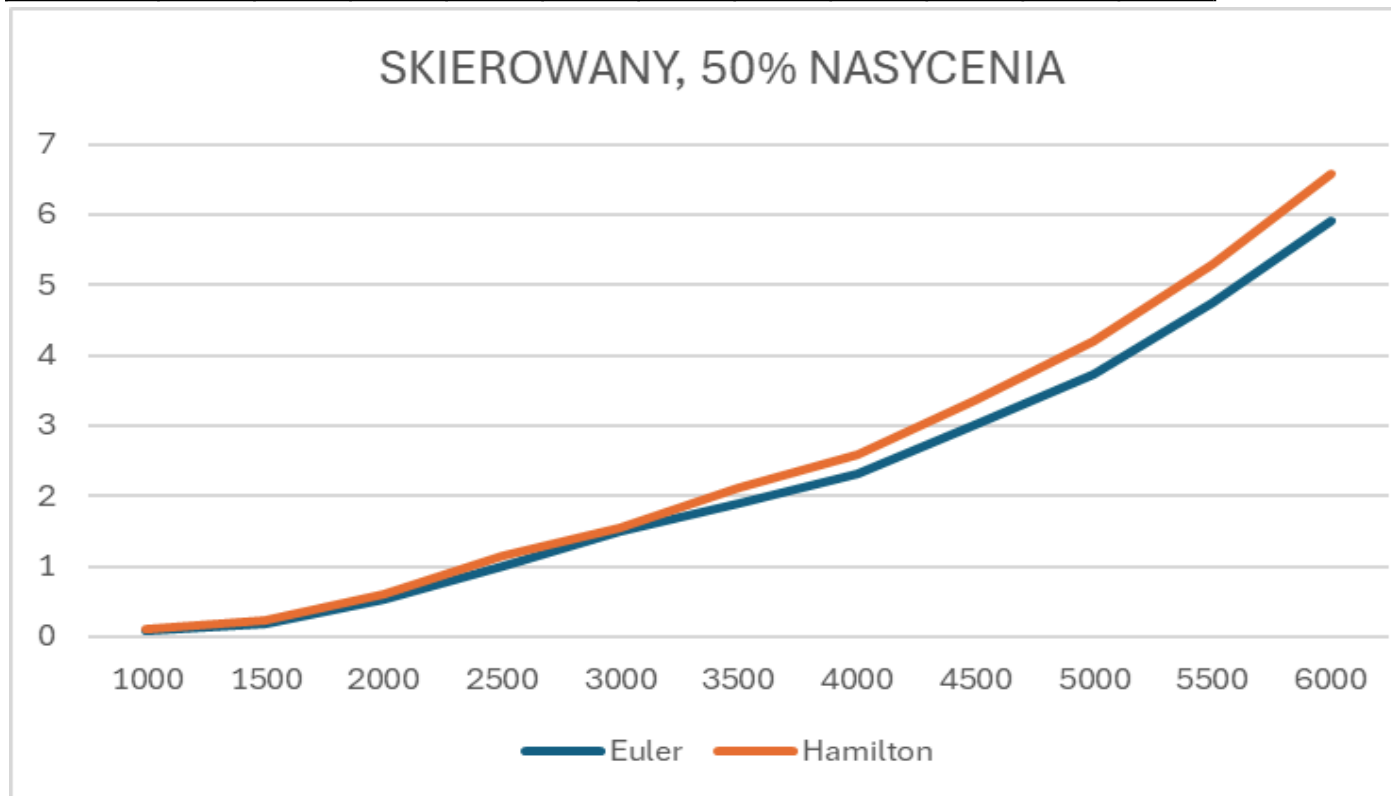
	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000
Graf Nieskierowany	0,08553	0,183655	0,579461	0,7912	1,1132	1,54573	1,9632	2,609	3,66228	4,70221	5,737
Graf Skierowany	0,112537	0,225534	0,602276	1,15018	1,43548	2,18874	2,50969	3,35195	4,20032	5,2972	6,57578



## GRAF SKIEROWANY

Zależność czasu obliczeń  $t$  od  $n$  liczby wierzchołków w grafie przy stałej wartości nasycenia  $s=50\%$ .

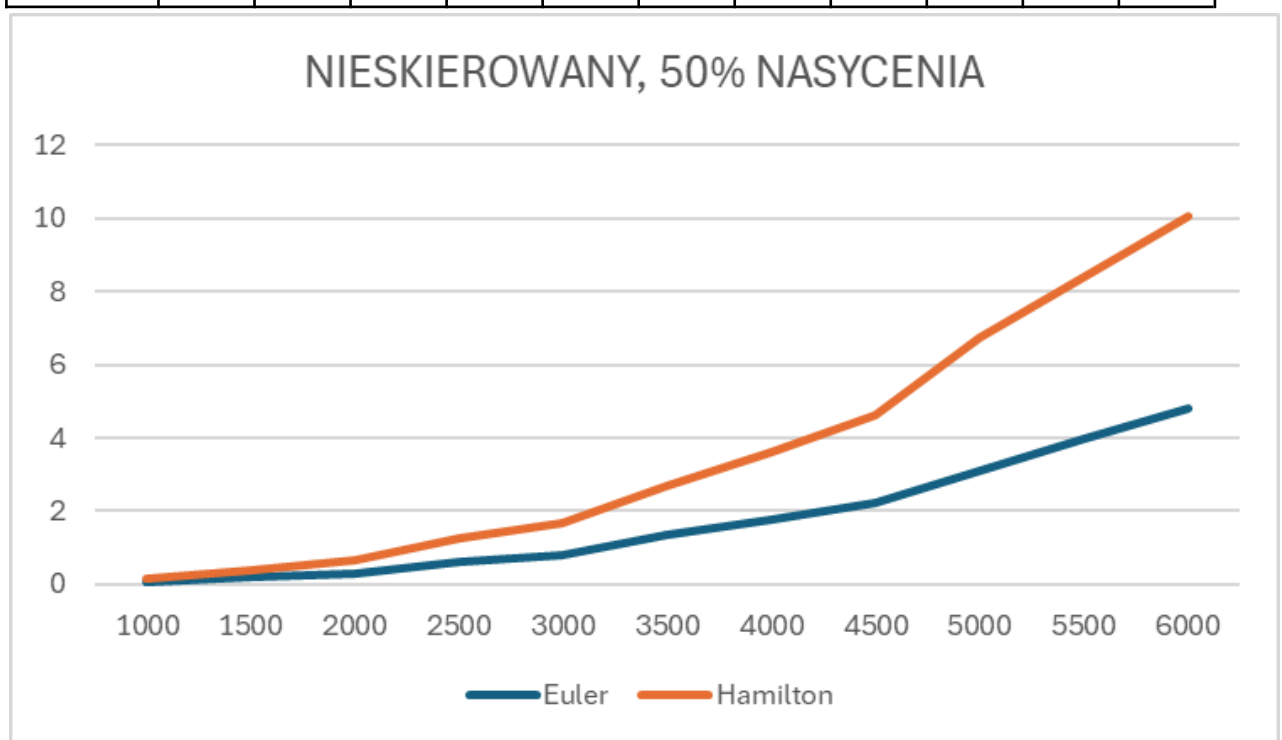
	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000
Euler	0,0812	0,1793	0,27512	0,62951	0,779	1,34331	1,7512	2,2129	3,1	3,991	4,8012
Hamilton	0,08553	0,183655	0,379461	0,6312	0,8832	1,34573	1,8632	2,409	3,66228	4,40221	5,237



## GRAF NIESKIEROWANY

Zależność czasu obliczeń  $t$  od  $n$  liczby wierzchołków w grafie przy stałej wartości nasycenia  $s=50\%$

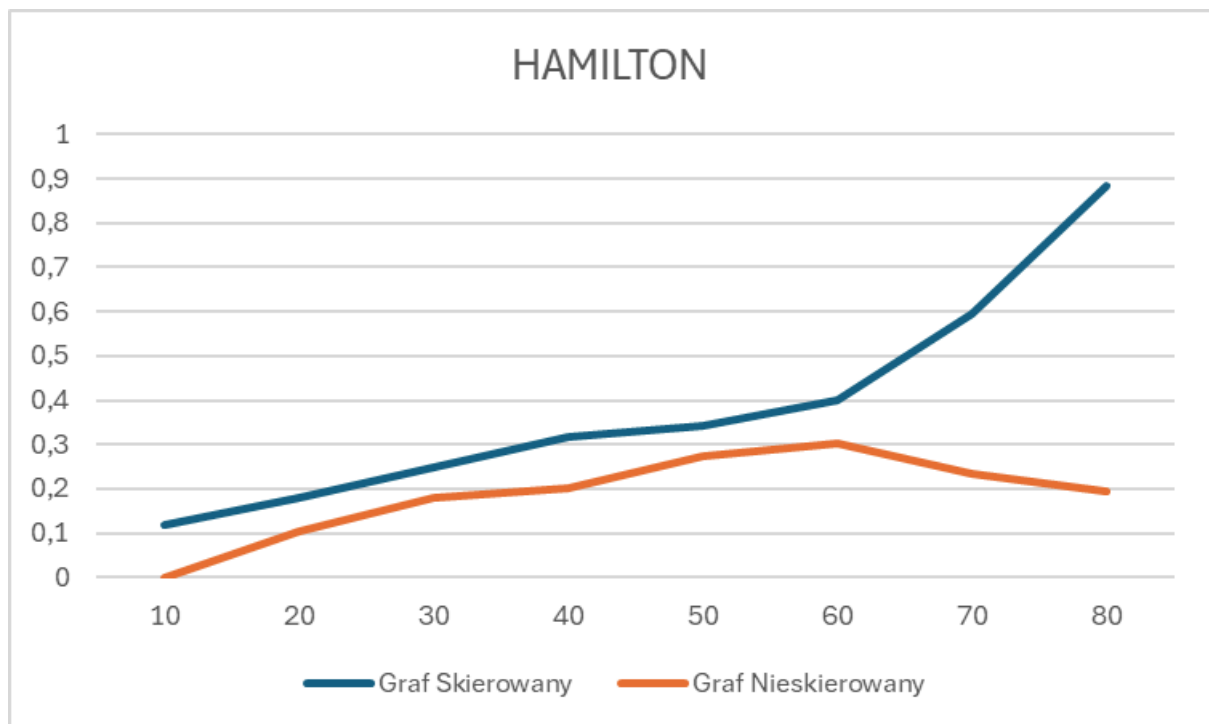
	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000
Euler	0,092771	0,18432	0,527276	1,00404	1,50448	1,90568	2,31512	3,01019	3,73973	4,74078	5,91105
Hamilton	0,112537	0,225534	0,602276	1,15018	1,53548	2,12174	2,60011	3,35195	4,20032	5,2972	6,57578



## CYKL HAMILTONA

Zależność czasu obliczeń  $t$  od liczby nasycenia wierzchołków w grafie.

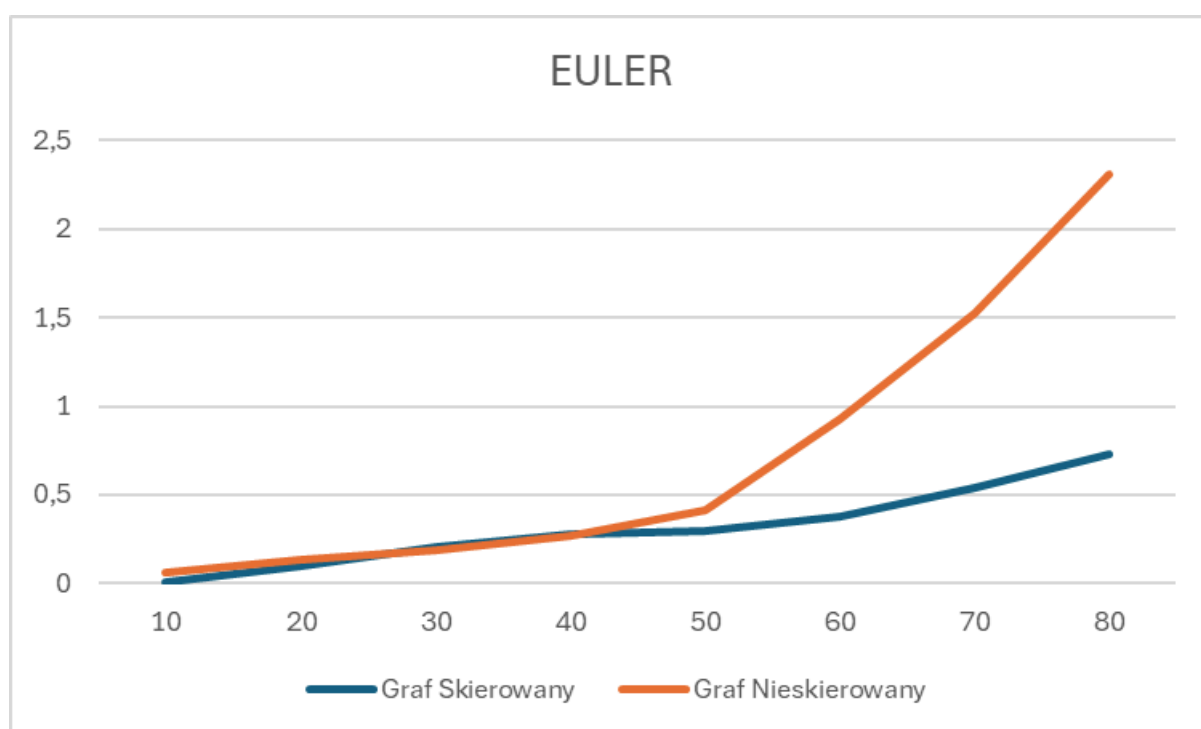
	10	20	30	40	50	60	70	80
Graf Skierowany	0,12011	0,180915	0,24713	0,317992	0,341199	0,40129	0,59636	0,883603
Graf Nieskierowany	0,001	0,1034	0,1794	0,2007	0,27512	0,301	0,2352	0,1935



## CYKL EULERA

Zależność czasu obliczeń  $t$  od liczby nasycenia wierzchołków w grafie.

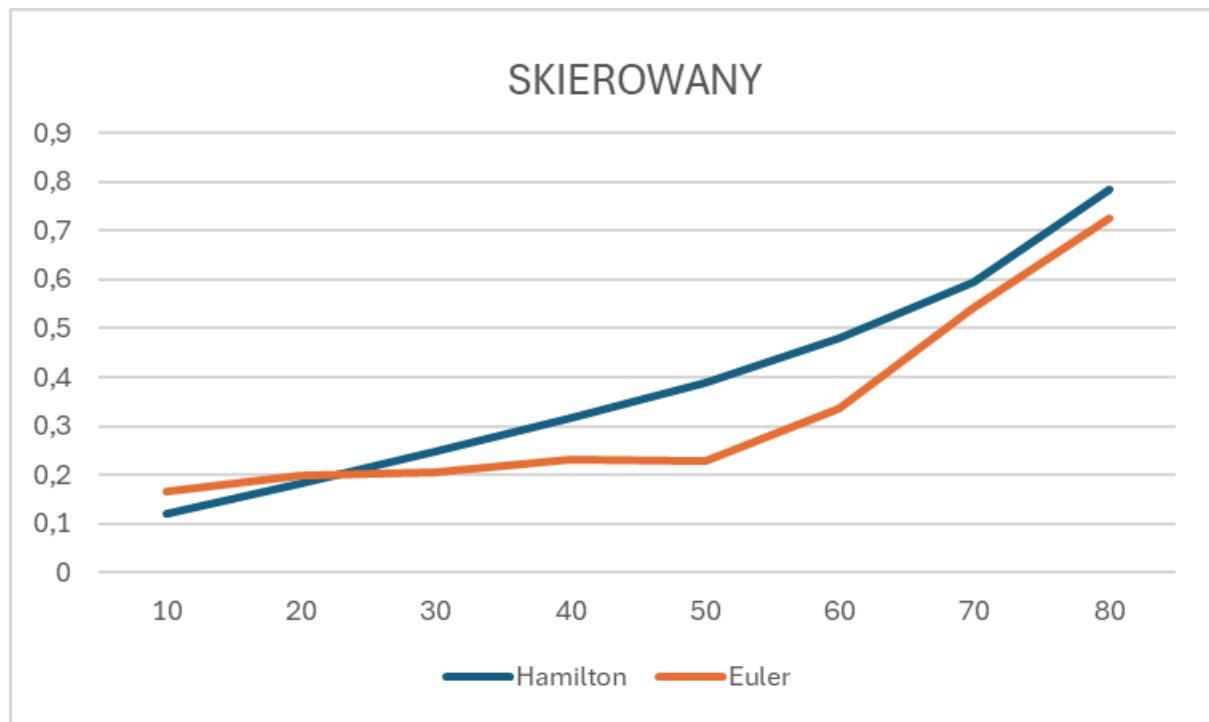
	10	20	30	40	50	60	70	80
Graf Skierowany	0,00436	0,096266	0,205238	0,280779	0,300406	0,375043	0,542054	0,727229
Graf Nieskierowany	0,06	0,1378	0,1854	0,2714	0,410461	0,9237	1,521	2,312



## GRAF SKIEROWANY

Zależność czasu obliczeń  $t$  od liczby nasycenia wierzchołków w grafie.

	10	20	30	40	50	60	70	80
Hamilton	0,12011	0,180915	0,24713	0,317992	0,388499	0,47899	0,59636	0,783603
Euler	0,16436	0,200066	0,205238	0,229779	0,229406	0,335043	0,542054	0,727229

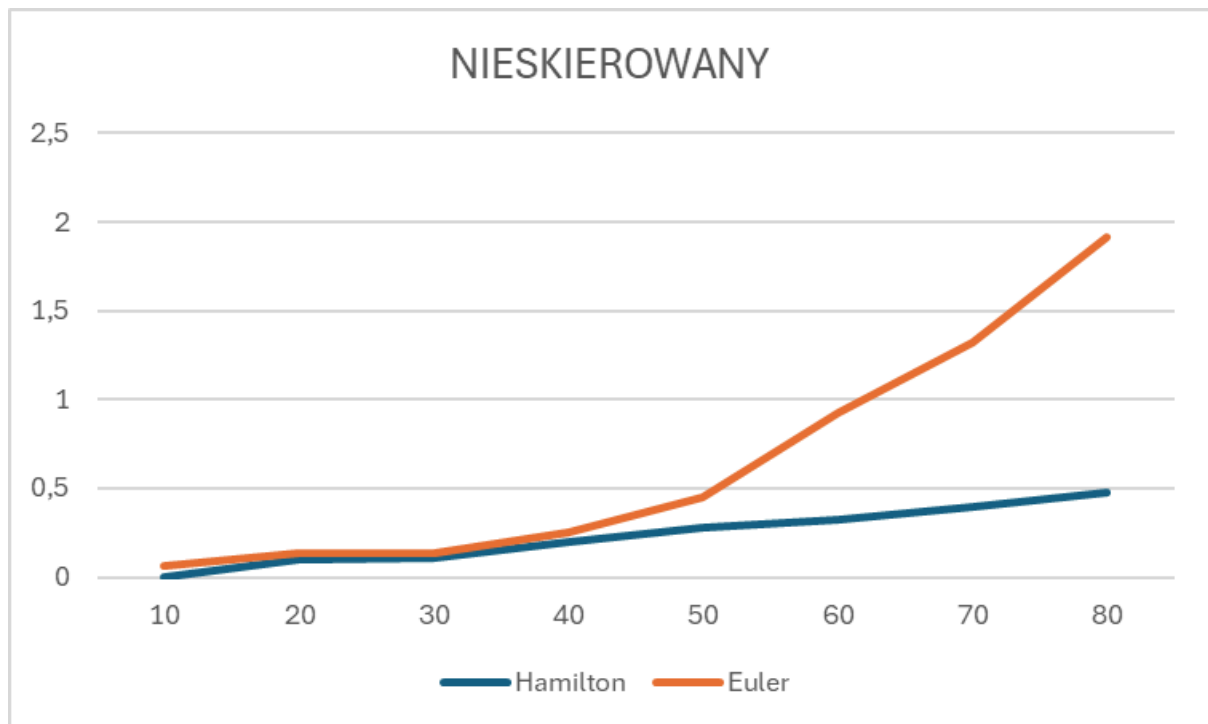


## GRAF NIESKIEROWANY

Zależność czasu obliczeń  $t$  od liczby nasycenia wierzchołków w grafie.

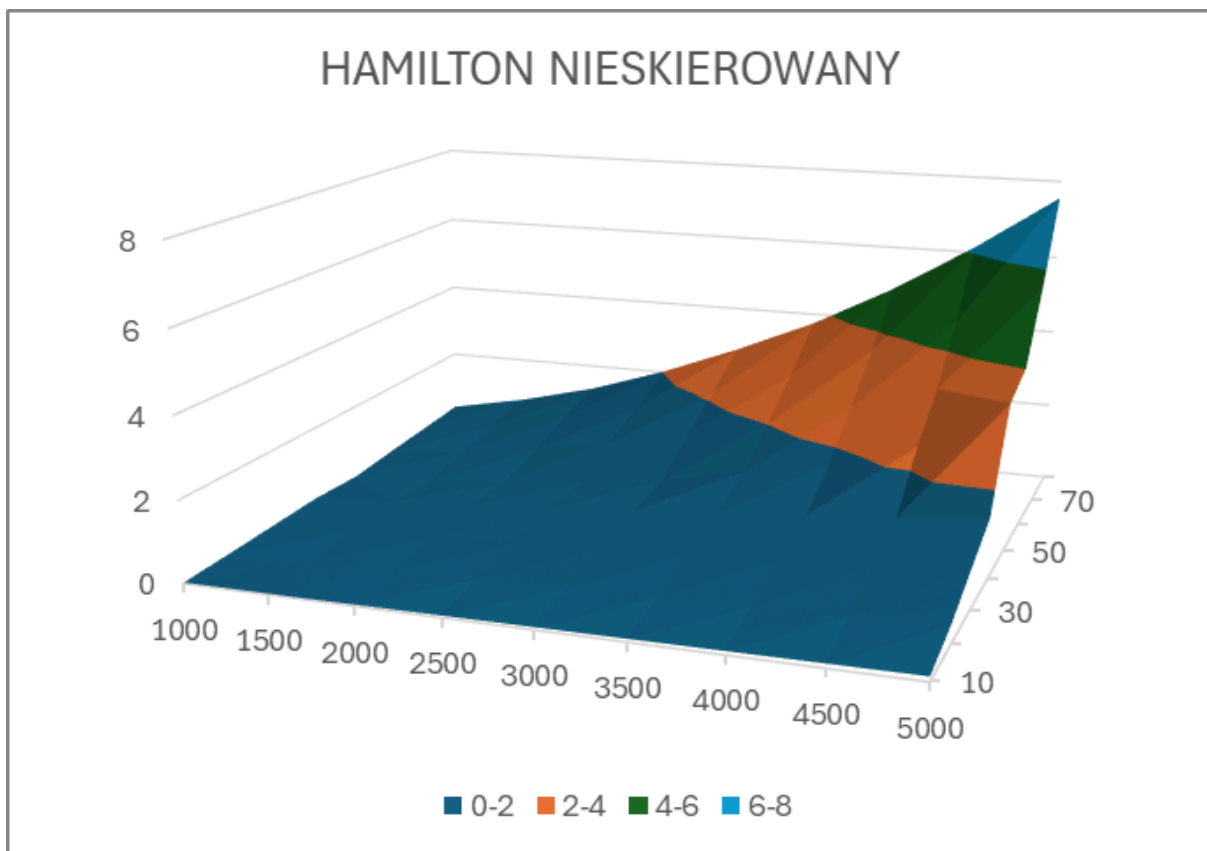
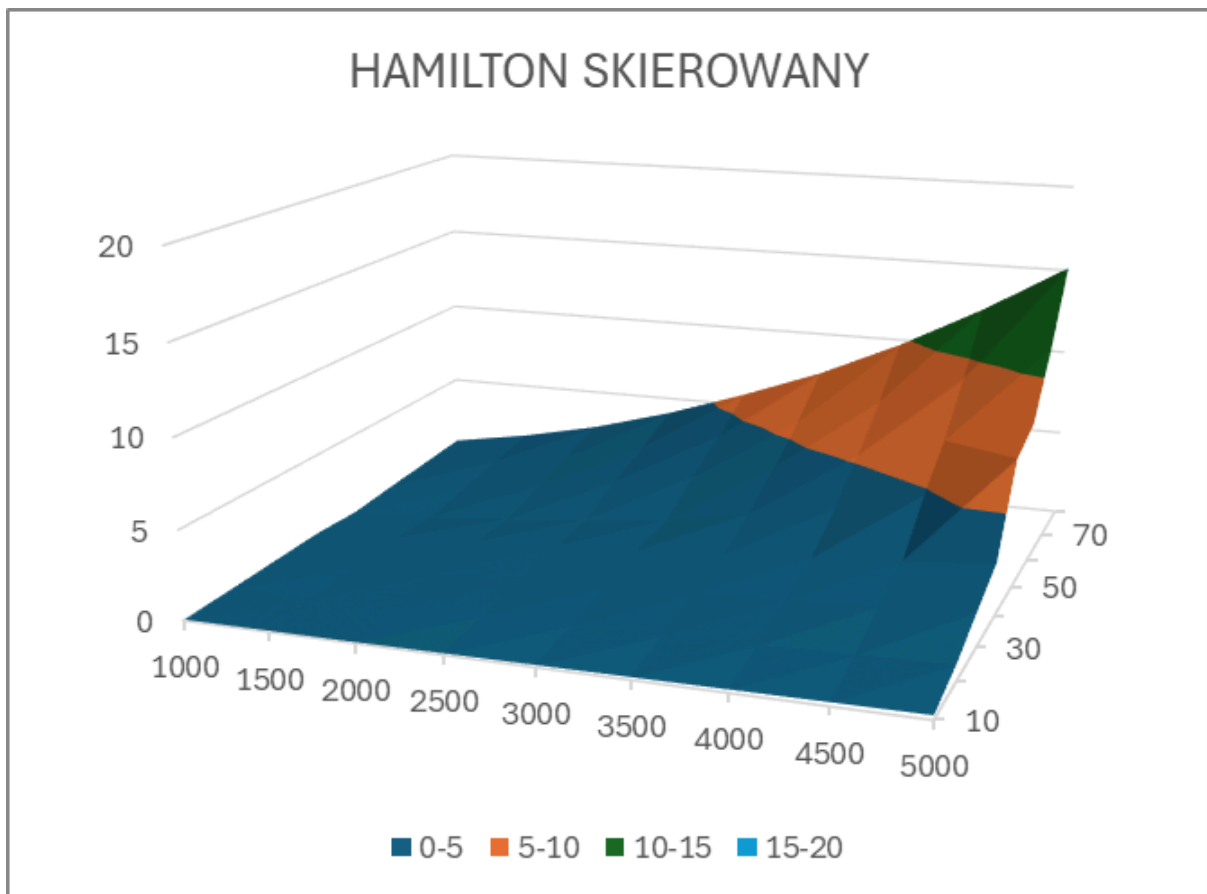


	10	20	30	40	50	60	70	80
Hamilton	0,001	0,1034	0,1094	0,2007	0,27512	0,321	0,3952	0,4735
Euler	0,06	0,1378	0,1354	0,2514	0,449461	0,9237	1,321	1,912



### WYKRES POWIERZCHNIOWY CYKL HAMILTONA

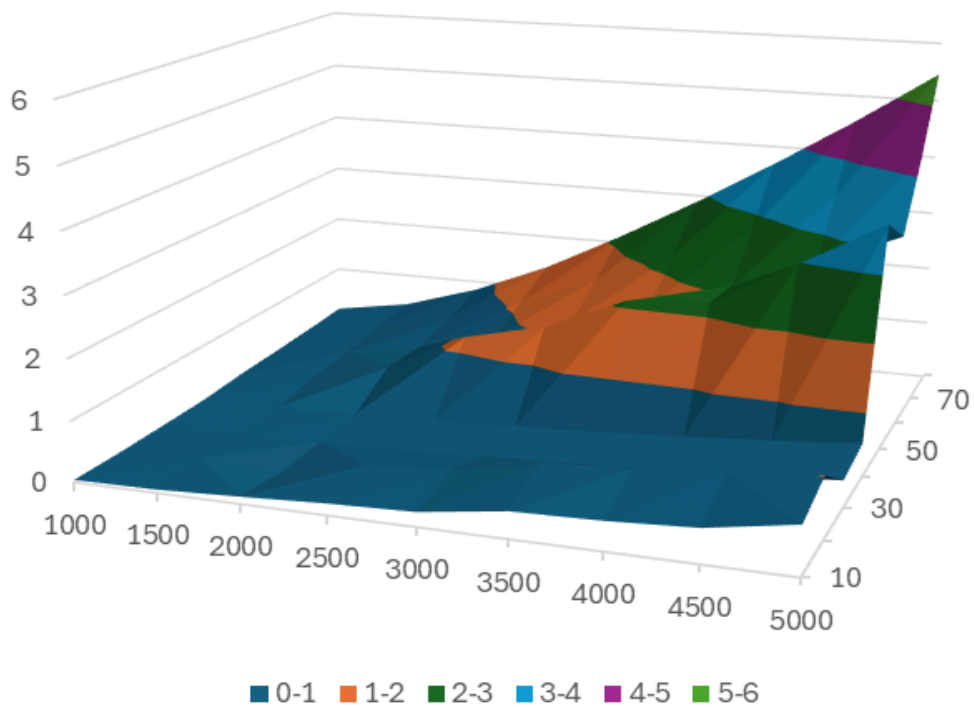
Zależność czasu  $t$  od liczby elementów i stopnia nasycenia wierzchołków.



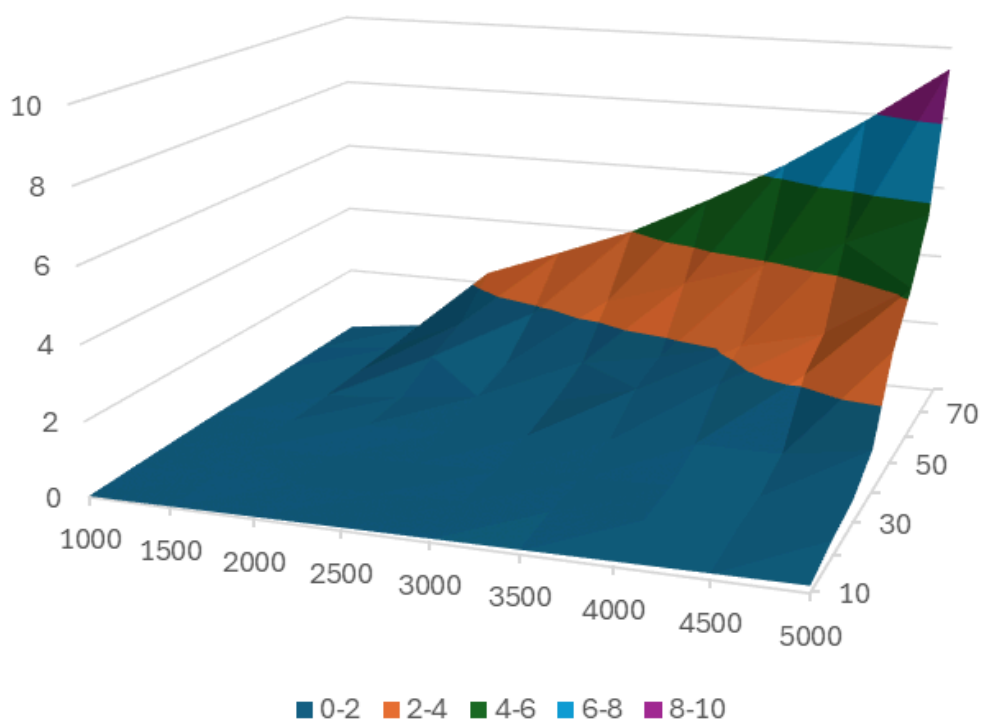
#### WYKRES POWIERZCHNIOWY CYKL EULERA

Zależność czasu  $t$  od liczby elementów i stopnia nasycenia wierzchołków.

## EULER SKIEROWANY



## EULER NIESKIEROWANY



## Sekcja II

Podsumowanie:

Złożoność obliczeniowa dla każdego z opisanych algorytmów składa się z dwóch składowych:

- złożoności przejścia przez graf
- znalezienia odpowiedniego wierzchołka (następnika)

Struktura danych	Sprawdzenie istnienia jednej krawędzi	Przejrzenie wszystkich sąsiadów wierzchołka*	Przejrzenie wszystkich krawędzi
Macierz sąsiedztwa	$O(1)$	$O(n)$	$O(n^2)$
Lista następników	$O(n)$	$O(n)$	$O(m)$

Dla cyklu hamiltona obie reprezentacje to  $O(V!)$ , natomiast dla cyklu eulera, macierz sąsiedztwa ma  $O(V^2)$ , a lista następników  $O(E+V)$ .

Algorytm znajdowania cyklu Hamiltona w grafie skierowanym zbudowanym za pomocą listy następników oraz w grafie nieskierowanym zbudowanym za pomocą macierzy sąsiedztwa różnią się pod względem struktury danych używanych do reprezentacji grafu. Każdy z tych podejść ma swoje wady i zalety. Oto główne przewagi algorytmu opartego na liście następników nad algorytmem opartym na macierzy sąsiedztwa:

### Przewagi algorytmu opartego na liście następników:

#### 1. Efektywność pamięciowa:

- **Lista następników:** Jest bardziej efektywna pamięciowo dla rzadkich grafów (grafów z niewielką liczbą krawędzi w porównaniu do liczby możliwych krawędzi). Dla grafu skierowanego z  $V$  wierzchołkami i  $E$  krawędziami, lista następników zajmuje  $O(V+E)$  pamięci.
- **Macierz sąsiedztwa:** Zajmuje  $O(V^2)$  pamięci niezależnie od liczby krawędzi. Dla dużych grafów o małej liczbie krawędzi macierz sąsiedztwa może być bardzo nieefektywna pamięciowo.

#### 2. Efektywność operacji:

- **Lista następników:** Przejście przez wszystkie sąsiednie wierzchołki danego wierzchołka zajmuje czas proporcjonalny do liczby sąsiednich wierzchołków, co jest korzystne w przypadku grafów rzadkich.
- **Macierz sąsiedztwa:** Przejście przez wszystkich sąsiadów wymaga sprawdzenia każdego wierzchołka, co zajmuje czas  $O(V)$ , nawet jeśli liczba sąsiednich wierzchołków jest mała.

#### 3. Czas inicjalizacji:

- **Lista następników:** Budowanie listy następników z danych wejściowych ma złożoność  $O(V+E)$ .
- **Macierz sąsiedztwa:** Inicjalizacja macierzy sąsiedztwa wymaga złożoności  $O(V^2)$ , co może być czasochłonne dla dużych grafów.

#### 4. Praktyczna użyteczność:

- **Lista następników:** Jest bardziej naturalną reprezentacją dla większości problemów grafowych, gdzie interesują nas bezpośrednie sąsiedztwa wierzchołków, szczególnie w przypadku grafów skierowanych.
- **Macierz sąsiedztwa:** Jest bardziej użyteczna w grafach gęstych, gdzie liczba krawędzi jest bliska maksymalnej możliwej liczbie krawędzi ( $V^2$ ) dla grafów nieskierowanych).

Wybór reprezentacji grafu zależy od specyfiki problemu i charakterystyki grafu (rzadki vs gęsty, skierowany vs nieskierowany).

Ze względu na swoją złożoność obliczeniową szukanie cyklu eulera należy do problemów:

P trudnych dla decyzyjnego.

NP trudnych dla przeszukiwania.

Ze względu na swoją złożoność obliczeniową szukanie hamiltona eulera należy do problemów NP - zupełnych dla obu reprezentacji.