



Politechnika Poznańska

Informatyka rok I semestr 2

L10, Piątek 11:45 - 13:15

Algorytmy i Struktury Danych

Prowadzący: Dominik Piotr Witczak

Sprawozdanie nr 4

Algorytmy z powracaniem

Autor:

Dominik Fischer 164176

Oliwer Miller 163544

Rok akademicki 2024/2025

Wprowadzenie

Celem niniejszego projektu było zaimplementowanie programu tworzącego i obsługującego grafy nieskierowane w dwóch postaciach:

- graf Hamiltonowski o wybranym nasyceniu (30% lub 70%),
- graf Nie-hamiltonowski o nasyceniu 50%.

Na danych grafach wykonywano operacje:

- wypisania grafu,
- wyszukiwania cyklu Eulera,
- wyszukiwania cyklu Hamiltona z użyciem algorytmu z powracaniem,
- eksportu grafu do formatu LaTeX.

Reprezentacja grafu

Jako reprezentację grafu wybraliśmy listę sąsiedztwa i zaimplementowaliśmy ją w formie słownika ze względu na:

- szybki dostęp do sąsiadów wierzchołka,
- łatwe dodawanie i usuwanie krawędzi,
- prostą implementację algorytmów przeszukiwania.

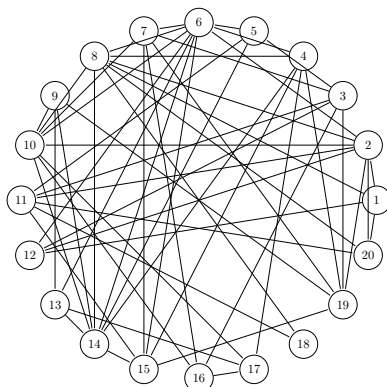
Terminal

```
class Graph:
def __init__(self, n):
self.n = n
self.adj = {i: set() for i in range(1, n + 1)}
```

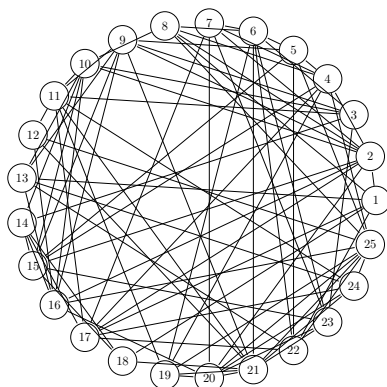
Wizualizacja utworzonych grafów

Grafy hamiltonowskie nieskierowane dla nasycenia wynoszącego 30%

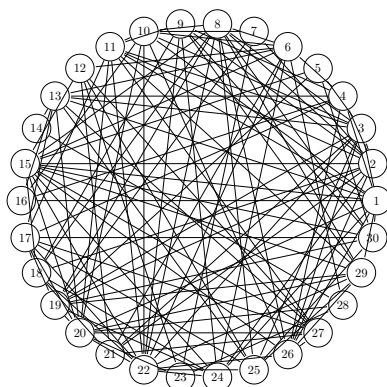
n = 20



n = 25

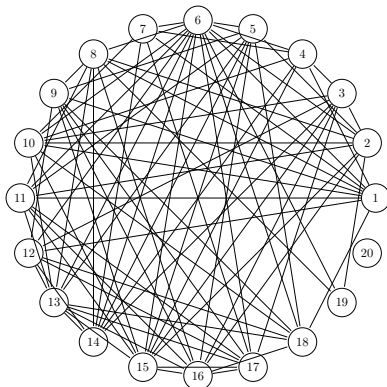


n = 30

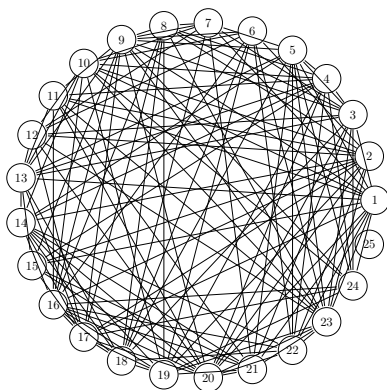


Grafy nie-hamiltonowskie nieskierowane dla nasycenia wynoszącego 50%

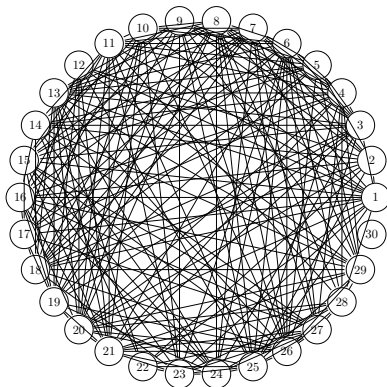
n = 20



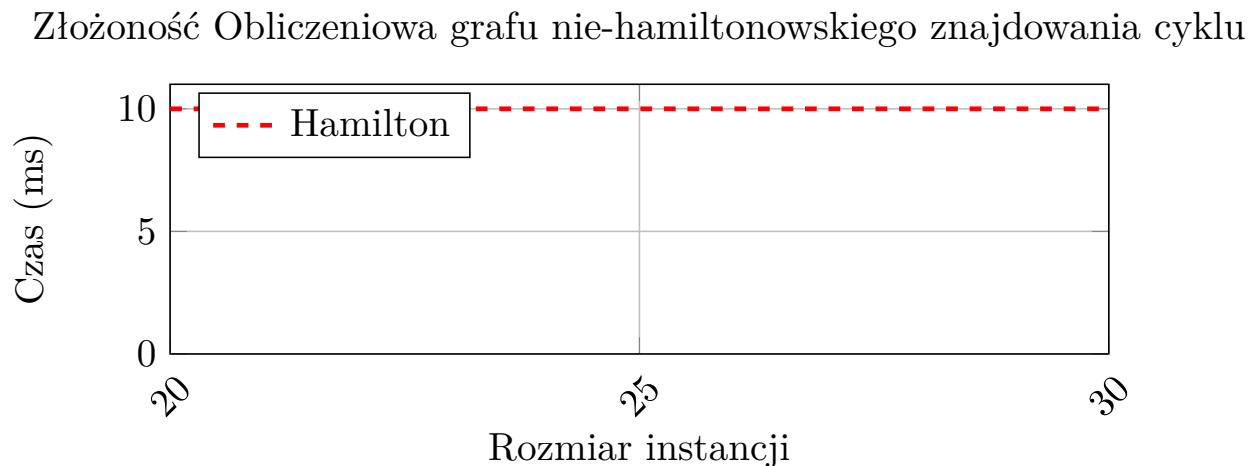
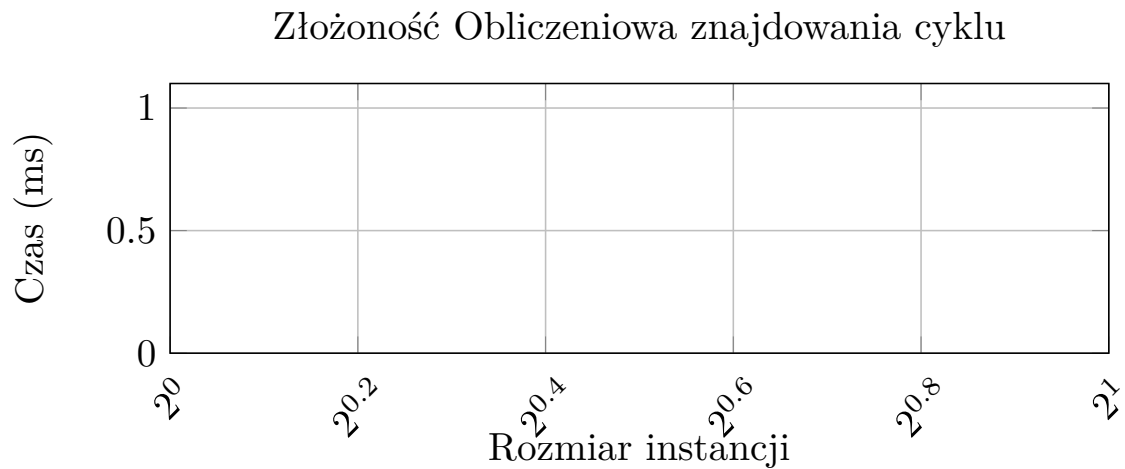
n = 25



n = 30



Porównanie czasów wykonania



Rysunek 1: Porównanie czasów wykonania dla 3 operacji

Wnioski

Na podstawie przedstawionych wykresów można wyciągnąć następujące wnioski:

- Algorytm znajdowania cyklu Eulera działa bardzo efektywnie – jego czas wykonania jest praktycznie zerowy niezależnie od rozmiaru instancji. Jest to zgodne z teorią, ponieważ problem znajdowania cyklu Eulera należy do klasy P i może być rozwiązany w czasie liniowym względem liczby krawędzi.
- Algorytm znajdowania cyklu Hamiltona wykazuje znacznie większą złożoność czasową. Czas wykonania silnie zależy od rozmiaru instancji i waha się w sposób nieregularny, co jest charakterystyczne dla problemów NP-trudnych. Dla większych instancji czas rośnie dynamicznie, co potwierdza jego praktyczną nieefektywność.
- W przypadku grafu nie-hamiltonowskiego, czas wykonania pozostaje stały niezależnie od rozmiaru instancji, co może oznaczać wcześniejsze zakończenie algorytmu po stwierdzeniu braku ścieżki Hamiltona.