Teoria Algorytmów i Obliczeń

Raport końcowy

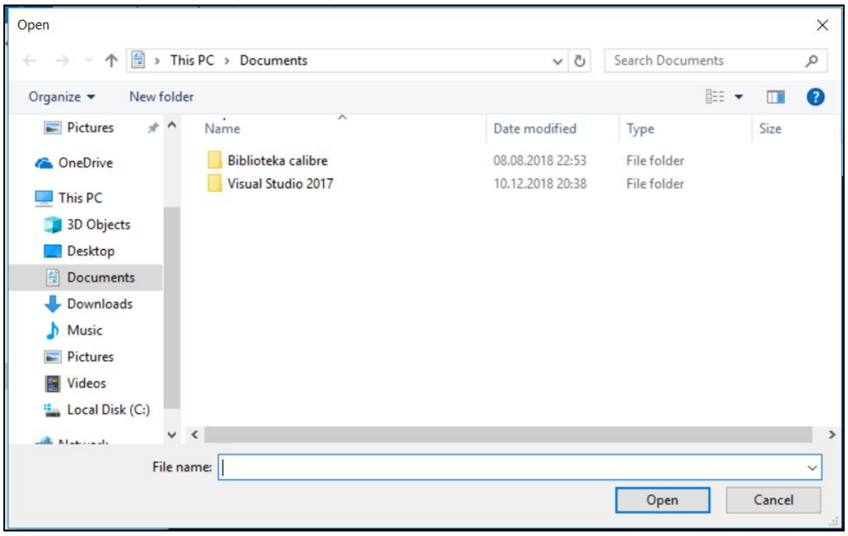
Aplikacja znajdująca maksymalną część wspólną dwóch spójnych grafów

Jakub Karolak
Konrad Kurach
Aleksy Miśtal

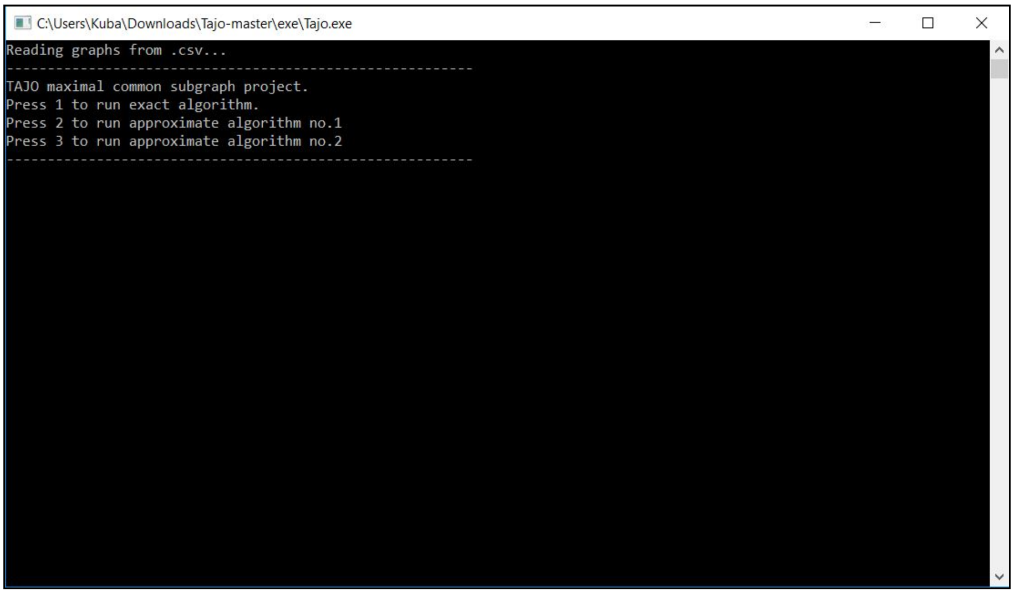
9 grudzień 2018

1. Uruchomienie aplikacji i weryfikacja wyników

Aby uruchomić program, należy przejść do katalogu ‘exe’, następnie uruchomić plik ‘Tajo.exe’. Program rozpocznie swoje działanie od uruchomienia okna wyboru grafu. Następnie należy nawigować o jeden poziom w górę i wybrać katalog ‘data’. W tym katalogu należy wybrać graf, który ma zostać użyty do wykonania algorytmów.



Po wybraniu grafu, pojawi się kolejne okno wyboru, dokładnie takie samo jak poprzednie. Należy wybrać drugi graf, na którym będzie operował program.



Jeśli oba grafy zostały dobrze wybrane, można przystąpić do części obliczeniowej programu. Aby uruchomić algorytm dokładny należy nacisnąć klawisz `1`, aby uruchomić algorytm aproksymacyjny należy nacisnąć klawisz `2` lub `3`, w zależności od tego czy naszą intencją jest uruchomienie pierwszego czy drugiego algorytmu aproksymacyjnego.

```
Reading graphs from .csv...
------------------------------------------------------------
TAJO maximal common subgraph project.
Press 1 to run exact algorithm.
Press 2 to run approximate algorithm no.1
Press 3 to run approximate algorithm no.2
------------------------------------------------------------
1
Press relevant key to choose exact alghoritm version:
 1) Vertices
 2) Vertices + Edges
```

Po wybraniu algorytmu dokładnego, należy jeszcze wybrać sposób liczenia. W przypadku naciśnięcia klawisza `1` algorytm wywoła się w wersji wierzchołkowej, w przypadku klawisza `2` w wersji wierzchołkowo-krawędziowej.

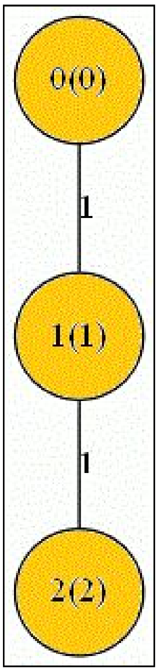
Po uruchomieniu algorytmu powinniśmy zobaczyć okno przeglądarki zawierające wizualizację części wspólnej grafów. Po wykonaniu obliczeń program wypisuje informacje o czasie wywołania.

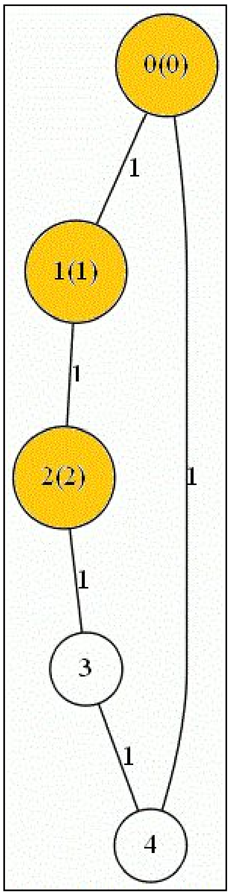
```
Exact algorithm - computing vertices...
0,1,2
0,1,2
Saving solution to C:\Users\Kuba\Downloads\Tajo-master\data\3_5_result1[1].csv
00:00:00.0185840
------------------------------------------------------------
Press [ESC] to exit...
Press [BACKSPACE] to load new graphs...
Press any other button to contiune calculations on given graphs...
```

Aby załadować nowe grafy do programu należy nacisnąć klawisz `Backspace`, aby wywołać inny algorytm na obecnych grafach, należy nacisnąć dowolny klawisz. Aby zakończyć działanie programu należy nacisnąć przycisk `Esc`.

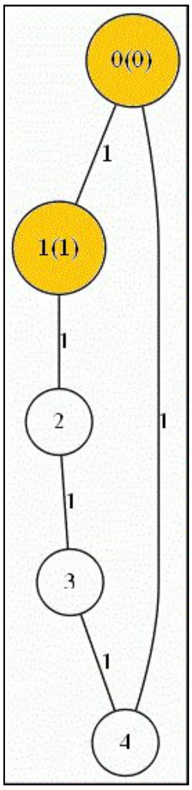
2. Prezentacja wyników algorytmu dokładnego

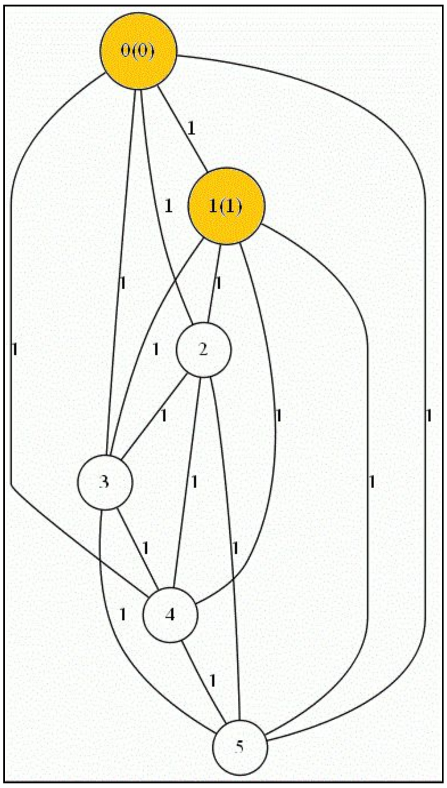
Graf pierwszy: 3 wierzchołki
Graf drugi: 5 wierzchołków
Czas wykonania: 00:00:00.0190011
Wersja: Ilość wierzchołków



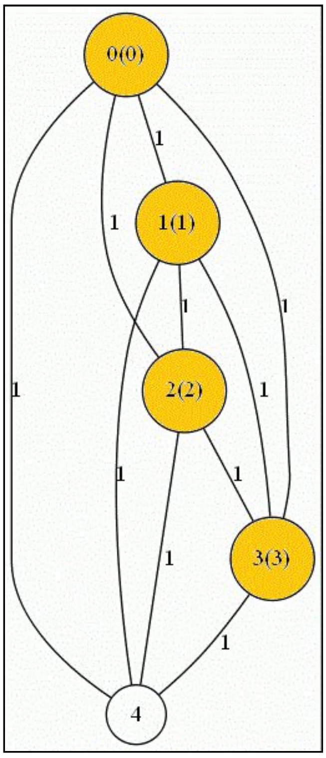


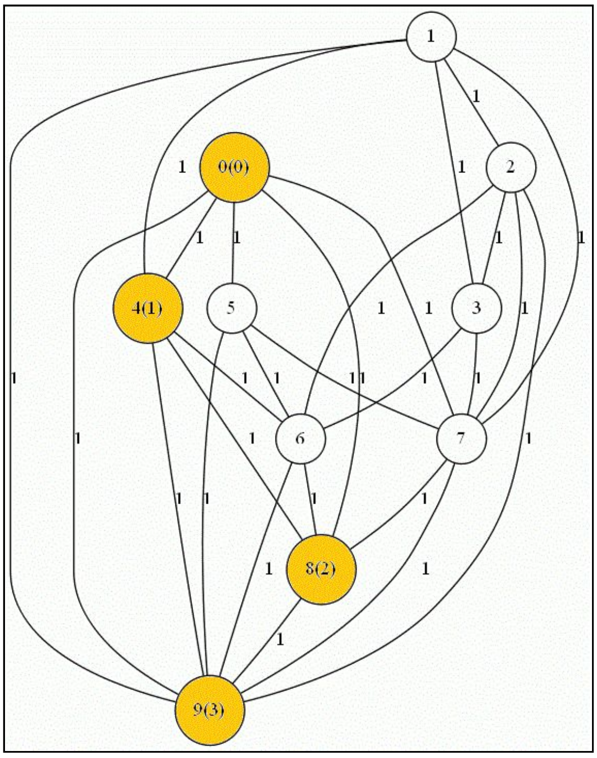
Graf pierwszy: 5 wierzchołków
Graf drugi: 6 wierzchołków
Czas wykonania: 00:00:00.0220013
Wersja: Ilość wierzchołków



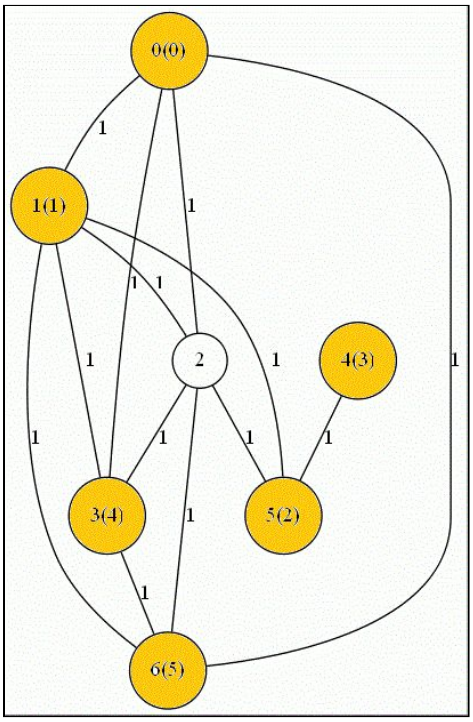


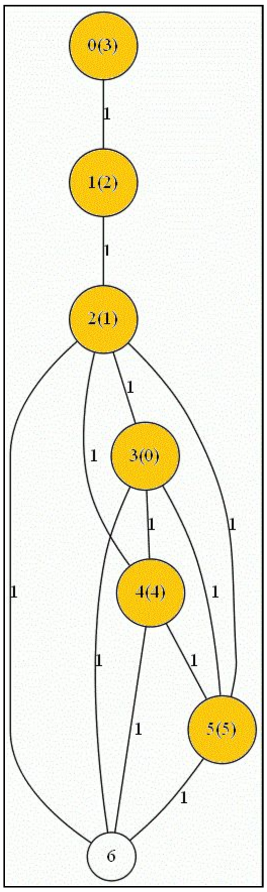
Graf pierwszy: 5 wierzchołków
Graf drugi: 10 wierzchołków
Czas wykonania: 00:00:00.0430025
Wersja: Ilość wierzchołków



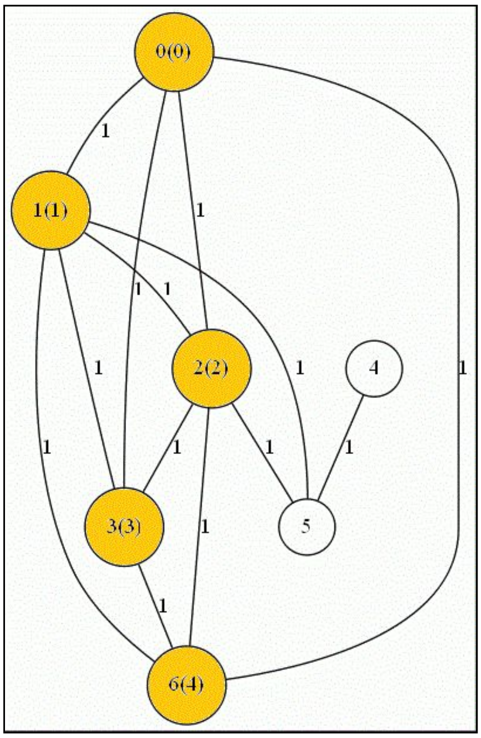


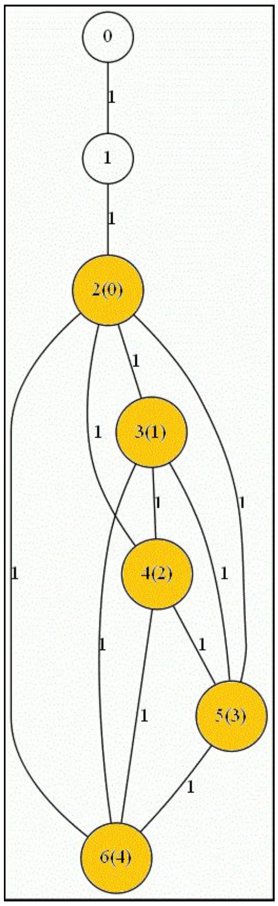
Graf pierwszy: 7 wierzchołków
Graf drugi: 7 wierzchołków
Czas wykonania: 00:00:00.0500028
Wersja: Ilość wierzchołków



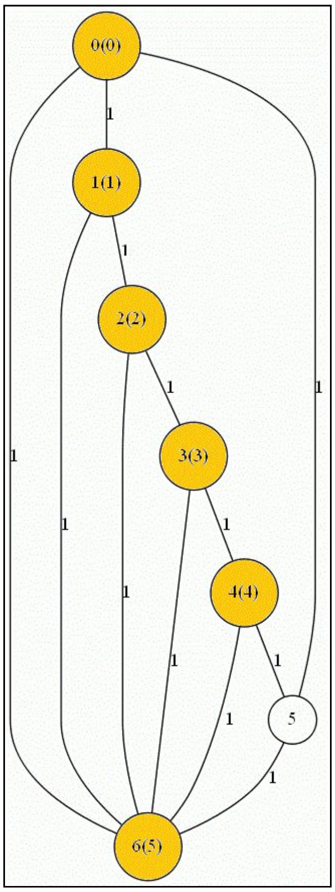


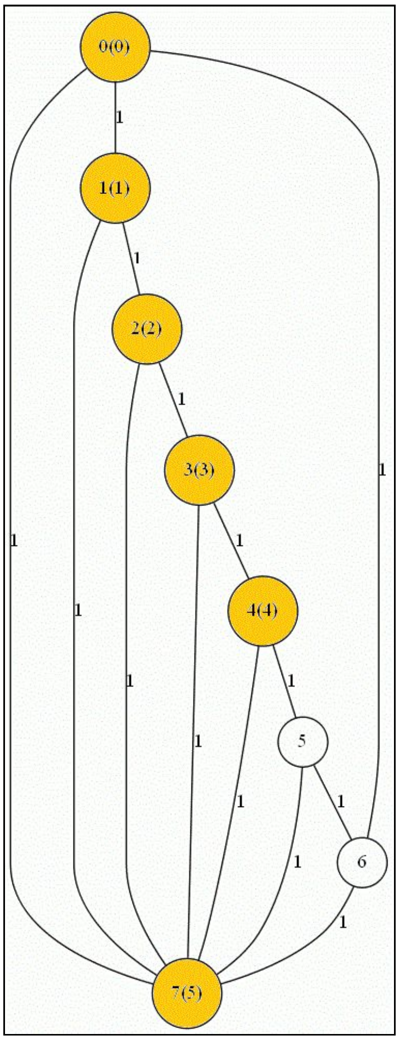
Graf pierwszy: 7 wierzchołków
Graf drugi: 7 wierzchołków
Czas wykonania: 00:00:00.0590034
Wersja: Ilość wierzchołków i krawędzi





Graf pierwszy: 7 wierzchołków
Graf drugi: 8 wierzchołków
Czas wykonania: 00:00:00.0300017
Wersja: Ilość wierzchołków



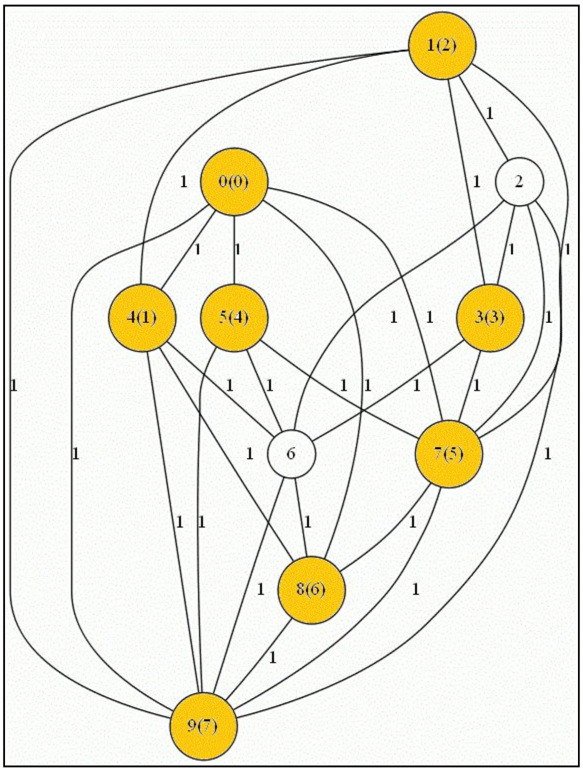


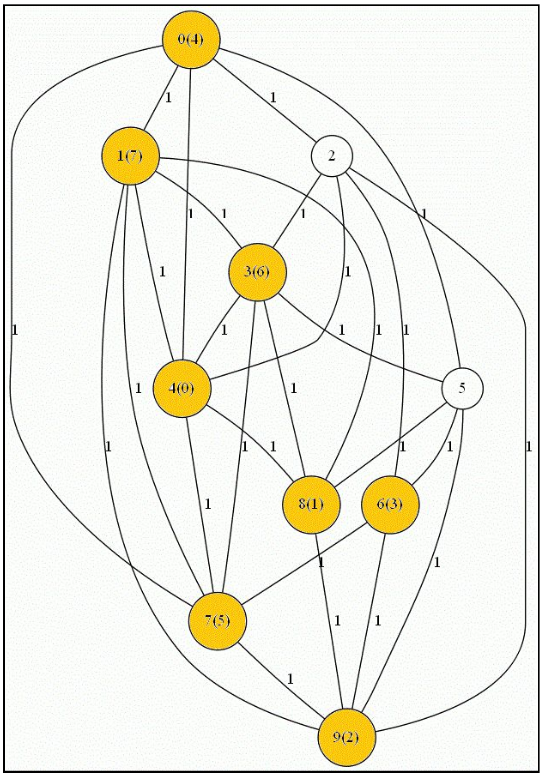
Graf pierwszy: 10 wierzchołków

Graf drugi: 10 wierzchołków

Czas wykonania: 00:00:00.5350306

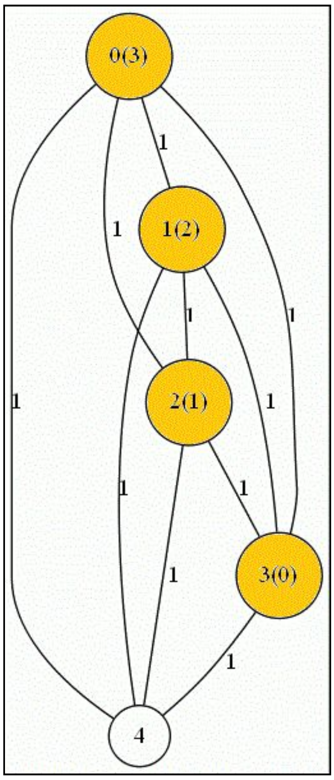
Wersja: Ilość wierzchołków

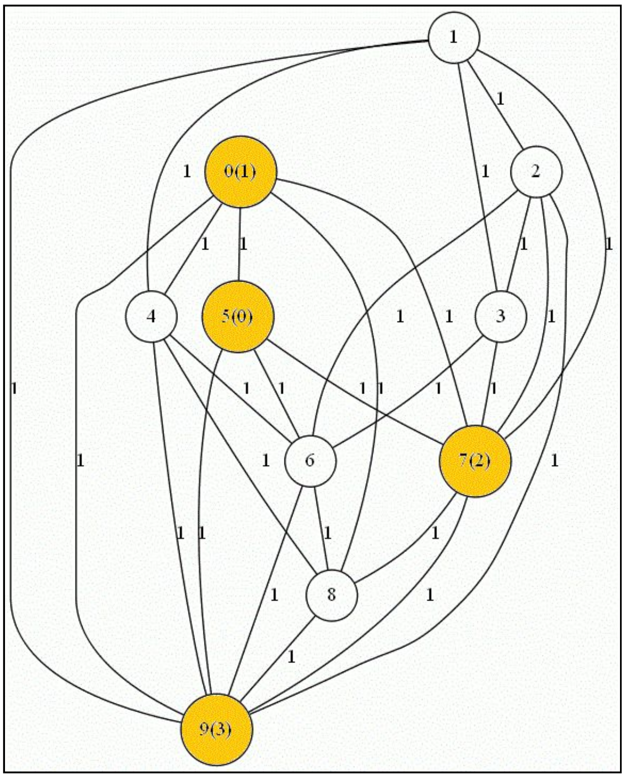




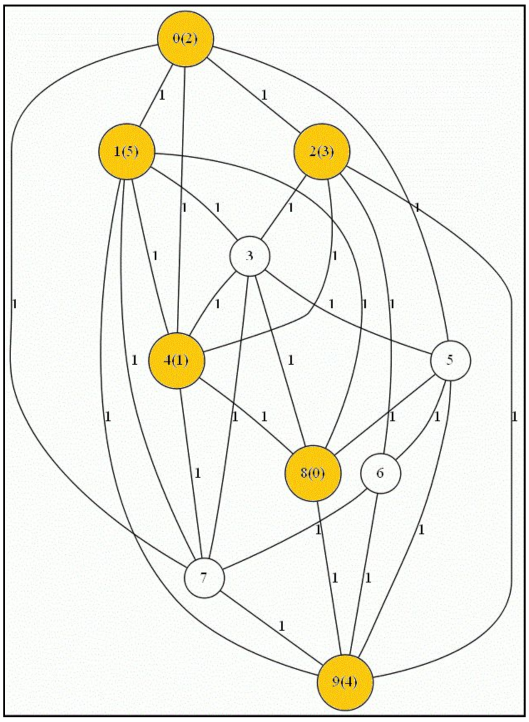
2. Prezentacja wyników algorytmu aproksymacyjnego 1

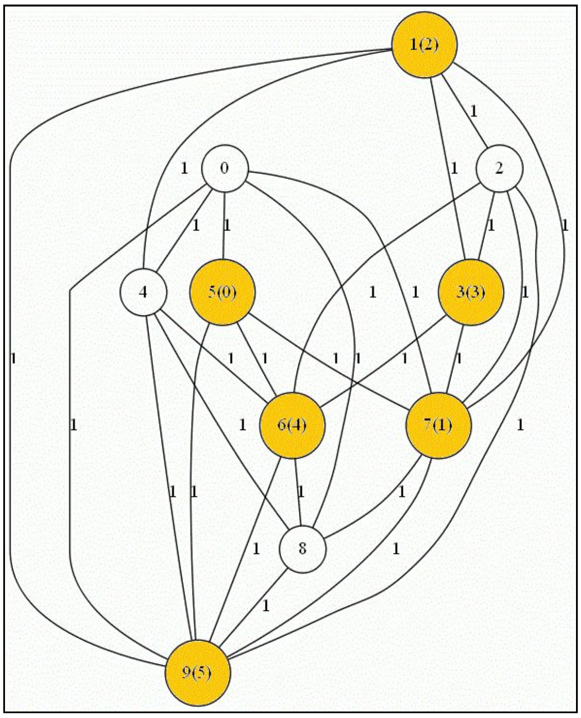
Graf pierwszy: 5 wierzchołków
Graf drugi: 10 wierzchołków
Czas wykonania: 00:00:00.0220013





Graf pierwszy: 10 wierzchołków
Graf drugi: 10 wierzchołków
Czas wykonania: 00:00:00.0230013

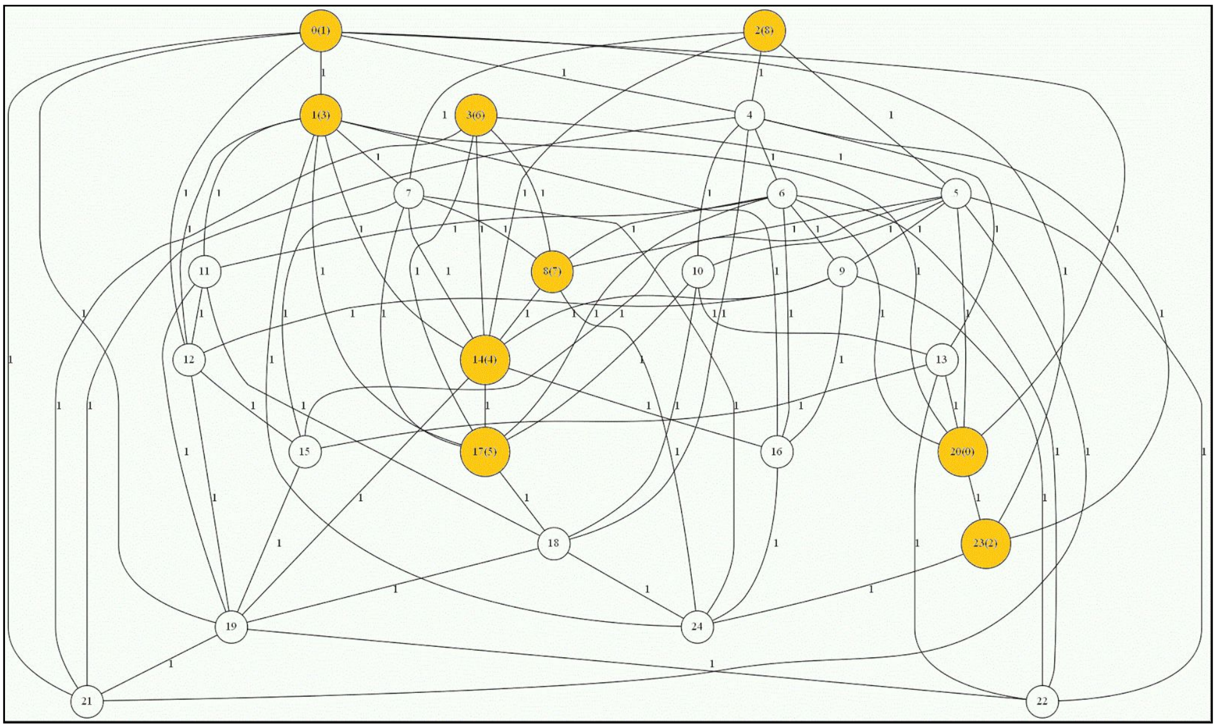


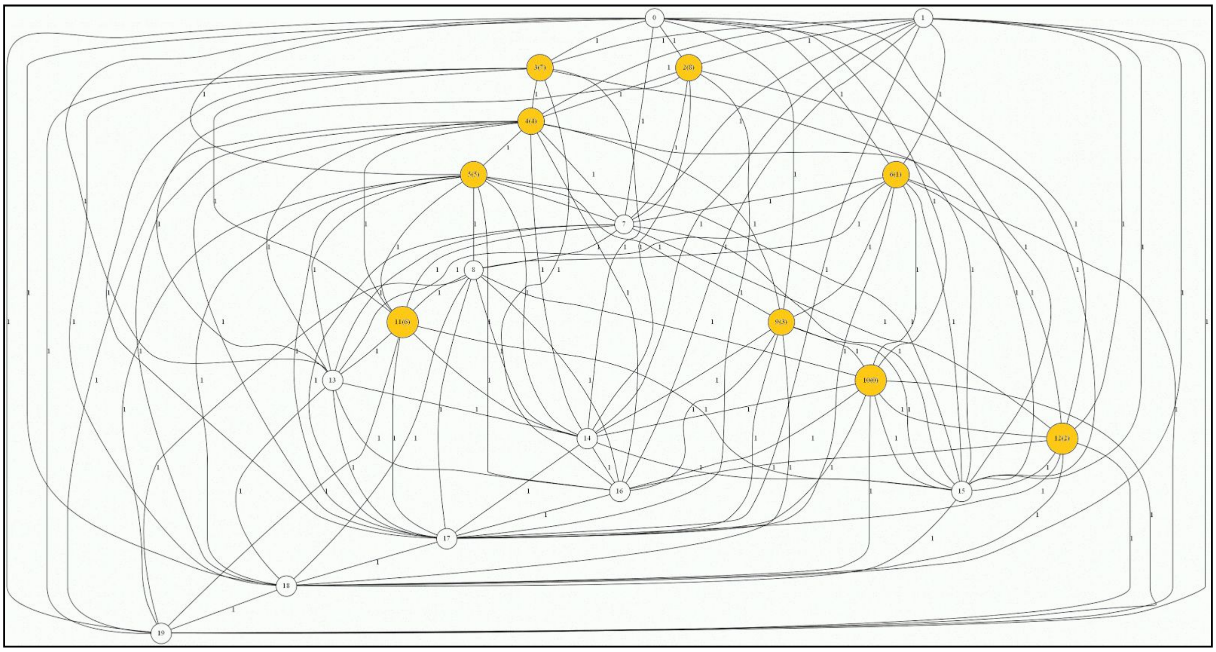


Graf pierwszy: 20 wierzchołków

Graf drugi: 25 wierzchołków

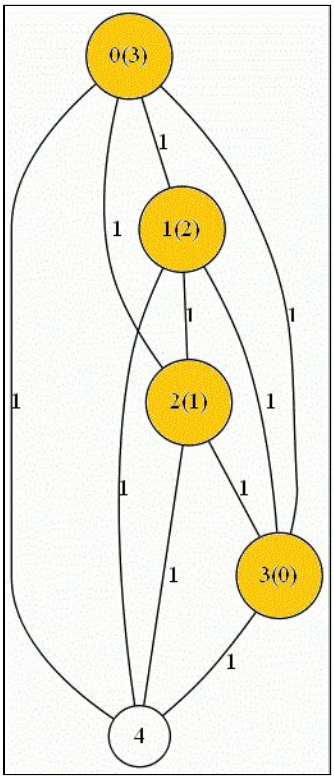
Czas wykonania: 00:00:00.0290017

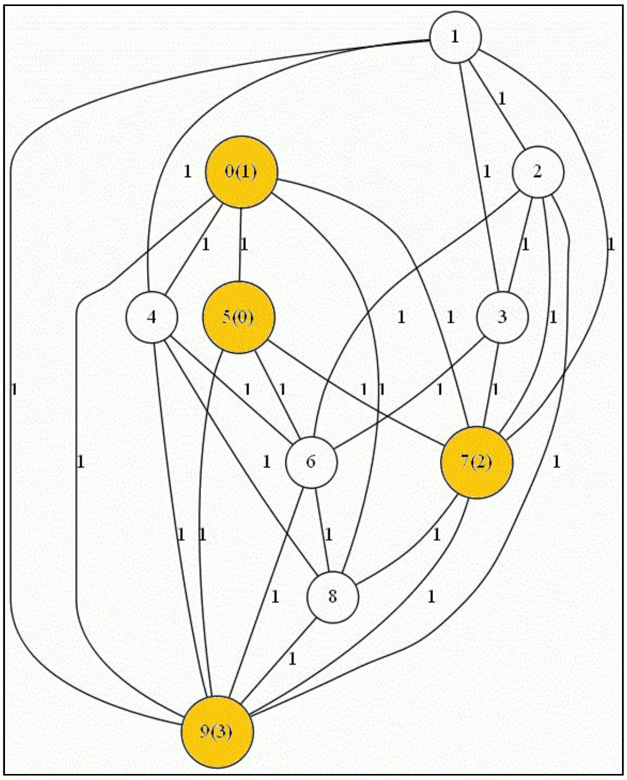




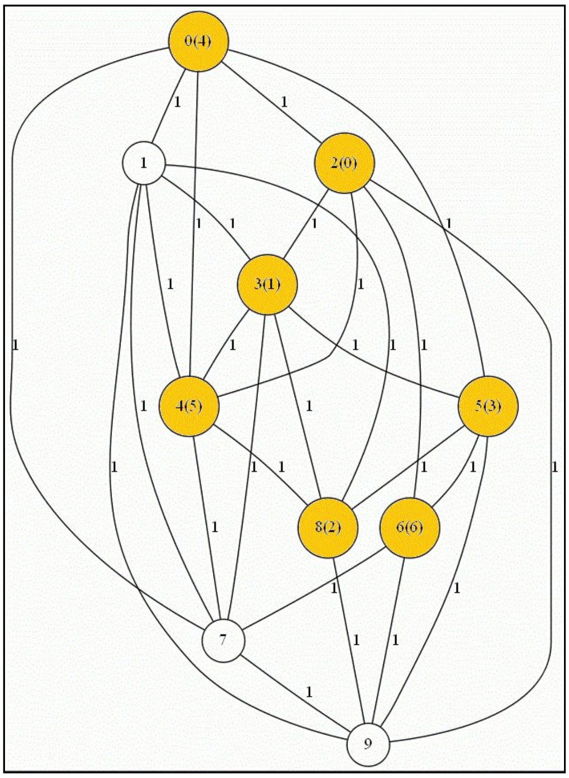
3. Prezentacja wyników algorytmu aproksymacyjnego 2

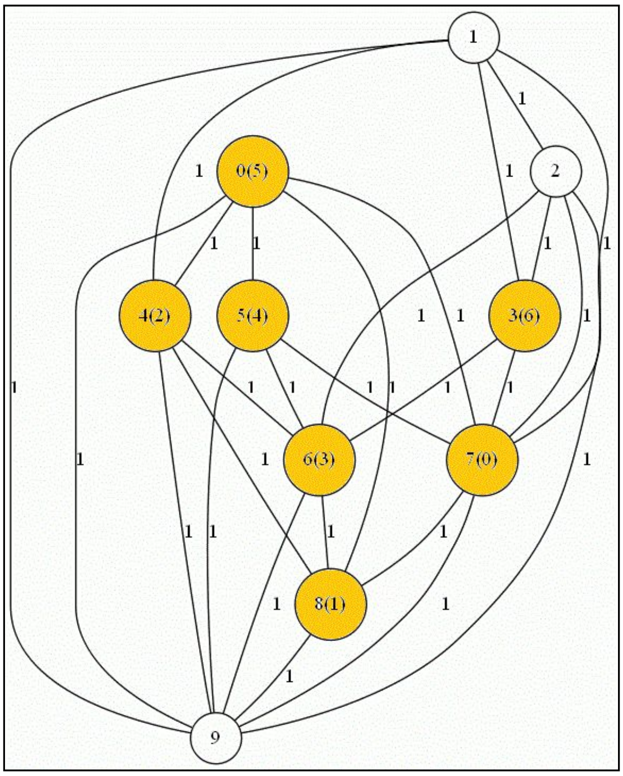
Graf pierwszy: 5 wierzchołków
Graf drugi: 10 wierzchołków
Czas wykonania: 00:00:00.0290017





Graf pierwszy: 10 wierzchołków
Graf drugi: 10 wierzchołków
Czas wykonania: 00:00:00.0350020

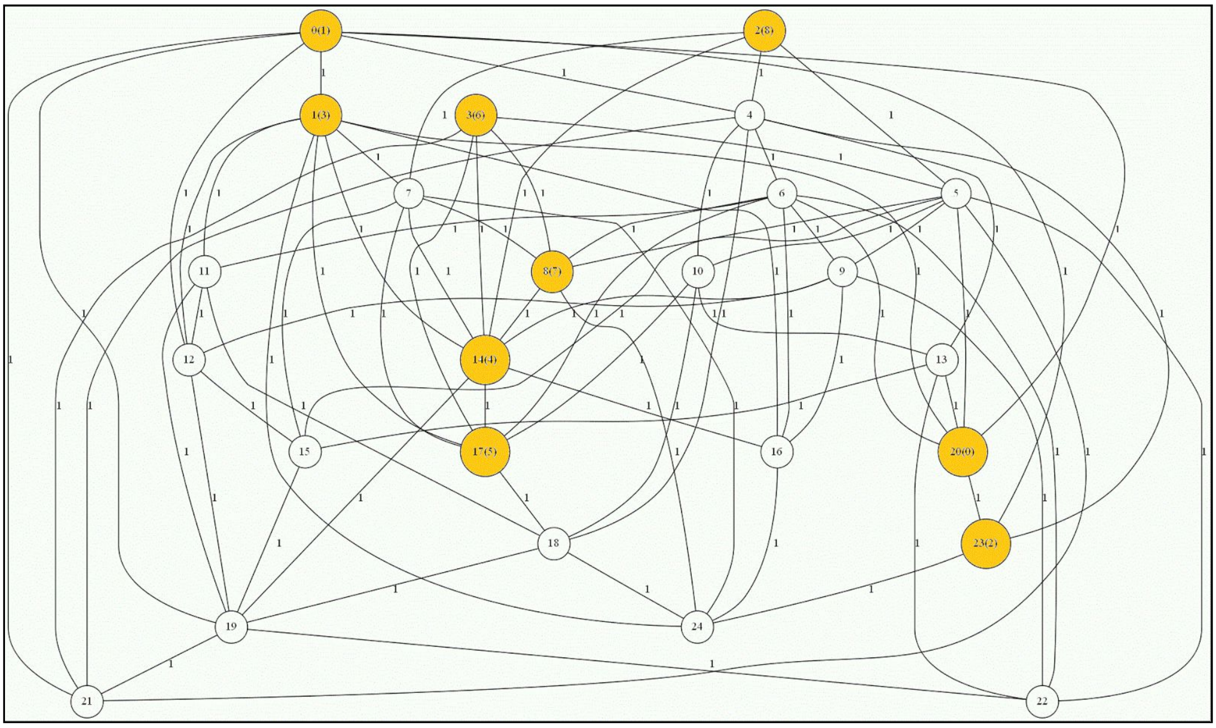


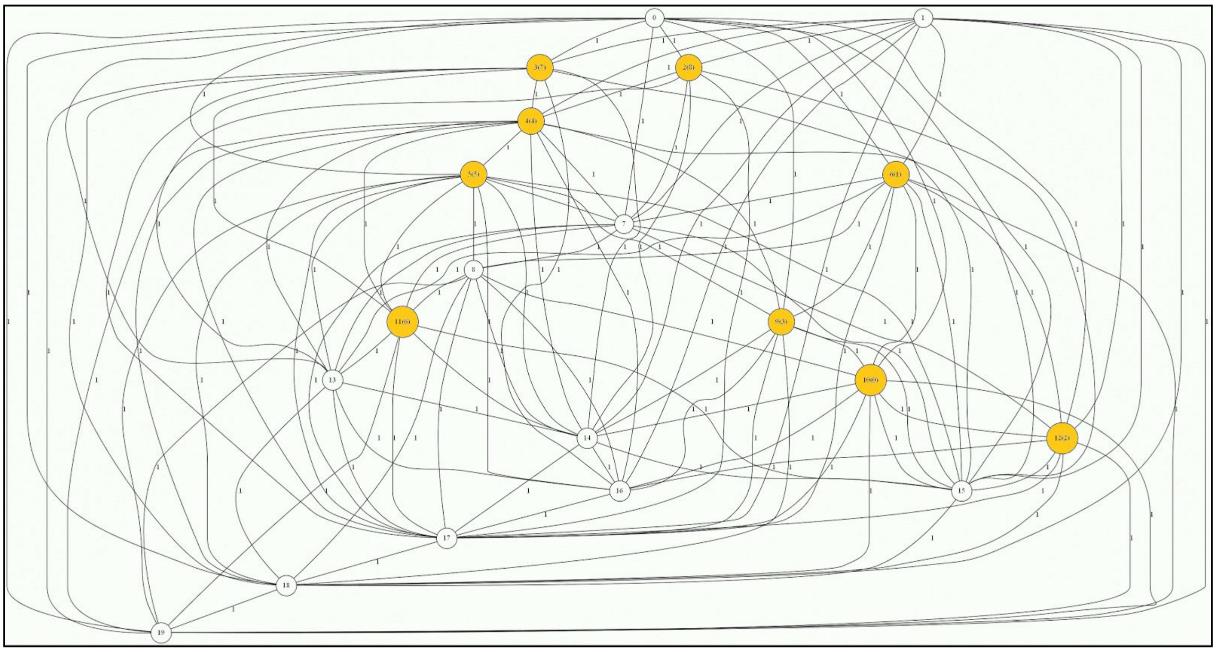


Graf pierwszy: 20 wierzchołków

Graf drugi: 25 wierzchołków

Czas wykonania: 00:00:01.7370994





4. Określenie złożoności algorytmu dokładnego

Czasowa złożoność obliczeniowa, była symulowana poprzez obliczenia algorytmu dla 50 par grafów o losowej gęstości oraz o równej liczbie wierzchołków.

Ponieważ w optymistycznym przypadku, algorytm potrafi znaleźć rozwiązanie w czasie wielomianowym, dlatego lepszym określeniem faktycznej złożoności jest ograniczenie górne, które wynosi $O(3^{n/3})$ gdzie:

m $-$ *ilość wierzchołków w pierwszym grafie*

c $-$ *ilość wierzchołków w drugim grafie*

$n = m * c$

W naszym przypadku:

$n = m^2$

Zaobserwowane przez nas wyniki wyniki bliższe są według równania krzywej trendu do $O(e^{1.6m})$. Której wykres przedstawiony jest poniżej.

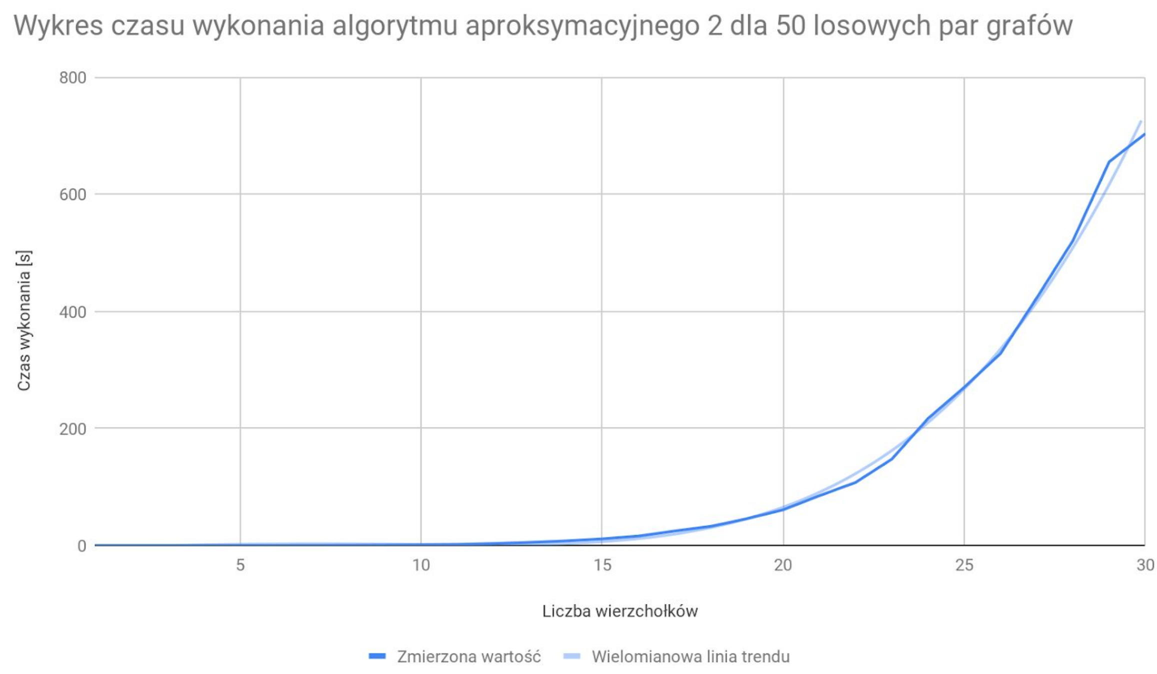
5. Określenie złożoności algorytmu aproksymacyjnego 1

Czasowa złożoność obliczeniowa, była również symulowana poprzez obliczenia algorytmu dla 50 par grafów o losowej gęstości oraz o równej liczbie wierzchołków.

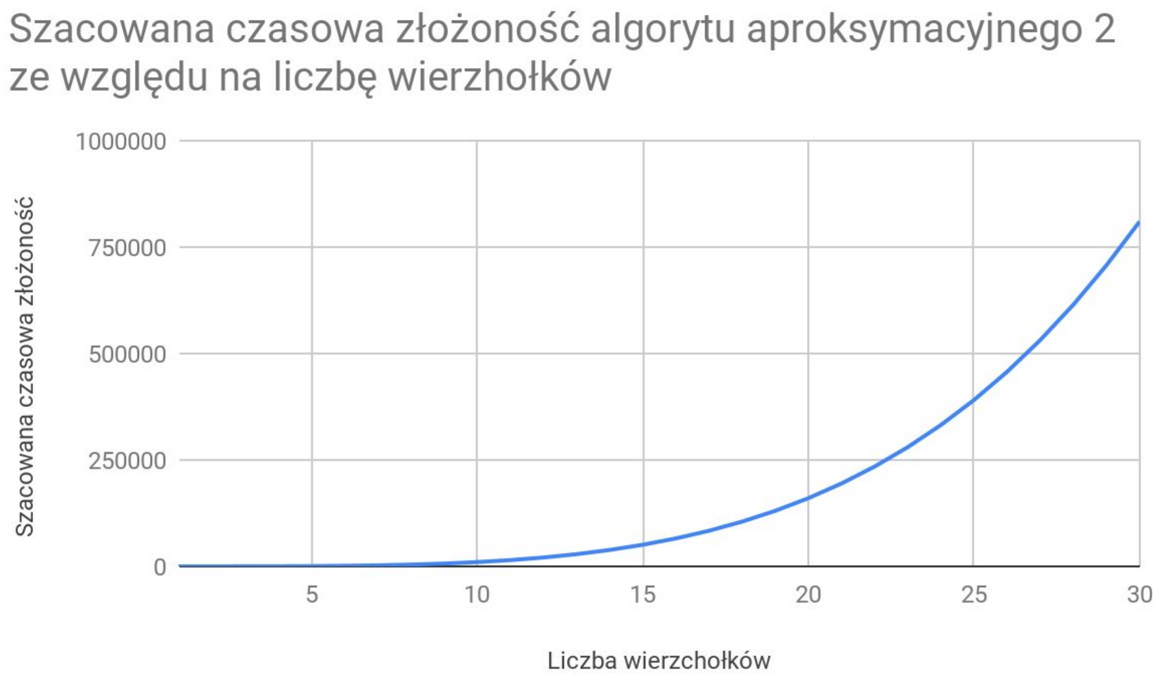
Jak widać na załączonym wykresie, nasze pomiary empiryczne dość dobrze pokrywają się dopiero z wielomianową linią trendu stopnia 3, czyli zgodnie z obliczoną przez nas złożonością ($O(n^3)$). Pokrycie jest wystarczająco dokładne, aby stwierdzić, że nasze oszacowanie złożoności algorytmu aproksymacyjnego 1 jest poprawne.

6. Określenie złożoności algorytmu aproksymacyjnego 2

Czasowa złożoność obliczeniowa, była symulowana poprzez obliczenia algorytmu dla 50 par grafów o losowej gęstości oraz o równej liczbie wierzchołków.



Jak widać na załączonym wykresie, nasze pomiary empiryczne dobrze pokrywają się z wielomianową linią trendu rzędu 4, co również jest zgodne z naszymi obliczeniami złożoności $O(n^4)$. Drobne różnice, pomiędzy szacowaną złożonością, a zaobserwowaną, które można zauważyć wynikają z losowości tworzonych grafów oraz niezbyt dużej próbki (50 par). Pokrycie jest wystarczająco dokładne, aby stwierdzić, że nasze oszacowanie złożoności algorytmu aproksymacyjnego 2 jest poprawne.



7. Wnioski

Biorąc pod uwagę wszystkie otrzymane wyniki, można wyciągnąć wniosek, iż estymaty złożoności zaimplementowanych algorytmów zostały wyliczone poprawnie, a program zachowywał się w sposób określony w dokumentacji. Algorytm dokładny, z racji swojej wykładniczej złożoności zwraca w akceptowalnym czasie wyniki dla grafów o liczebności wierzchołków co najwyżej 10. Dla większej ilości wierzchołków czasy wykonania rosną już bardzo szybko (godziny, dni, lata). Algorytmy aproksymacyjne są dokładniejsze dla grafów gęstych, a mniej dokładne dla grafów rzadkich, co spowodowane jest wyborem wierzchołka o najwyższym stopniu. Podsumowując, projekt ten został zakończony sukcesem.