Sprawozdanie – Laboratoria 4.

Karol Wziątek

20 stycznia 2026

1 Wprowadzenie

Na tej liście należy zapoznać się z problemem `max flow`, czyli znalezieniu maksymalnego przepływu w sieci przepływowej.

2 Algorytmy - implementacja

2.1 Metoda Forda–Fulkersona

Metoda Forda–Fulkersona jest ogólną procedurą wyznaczania maksymalnego przepływu w sieci przepływowej $G = (V, E)$ z wyróżnionym źródłem s, ujściem t oraz pojemnościami $c(u, v) \geq 0$. Idea metody polega na wielokrotnym znajdowaniu *ścieżek powiększających* w grafie rezydualnym i zwiększaniu przepływu wzdłuż tych ścieżek, aż do momentu, gdy dalsze powiększanie nie jest możliwe.

Graf rezydualny. Dla aktualnego przepływu f definiuje się pojemność rezydualną

$$c_f(u, v) = c(u, v) - f(u, v),$$

czyli ile dodatkowego przepływu można jeszcze przesłać krawędzią (u, v). Aby umożliwić modyfikację wcześniej wysłanego przepływu, do grafu rezydualnego dodaje się także krawędzie wsteczne: jeśli $f(u, v) > 0$, to w G_f istnieje krawędź (v, u) o pojemności $c_f(v, u) = f(u, v)$, pozwalająca „cofnąć" część przepływu. Graf rezydualny G_f składa się z tych krawędzi (u, v), dla których $c_f(u, v) > 0$.

Ścieżka powiększająca i augmentacja. Ścieżka P od s do t w grafie rezydualnym G_f nazywana jest ścieżką powiększającą. Dla takiej ścieżki wyznacza się *wąskie gardło*:

$$\Delta = \min_{(u,v)\in P} c_f(u, v),$$

czyli maksymalną wartość, o jaką można zwiększyć przepływ wzdłuż całej ścieżki, nie przekraczając pojemności. Następnie aktualizuje się przepływ:

- jeśli krawędź (u, v) na ścieżce jest zgodna z kierunkiem oryginalnej krawędzi, zwiększa się $f(u, v)$ o Δ,
- jeśli wykorzystano krawędź wsteczną (v, u), to zmniejsza się $f(u, v)$ o Δ (co odpowiada cofnięciu części przepływu).

Po tej operacji zmieniają się pojemności rezydualne i w konsekwencji sam graf rezydualny G_f.

Warunek zakończenia. Procedura jest powtarzana, dopóki istnieje ścieżka powiększająca z s do t w grafie rezydualnym. Gdy t staje się nieosiągalne z s w G_f, nie ma już możliwości powiększenia przepływu i uzyskany przepływ jest maksymalny. Z twierdzenia max-flow min-cut wynika, że brak ścieżki powiększającej jest równoważny istnieniu przekroju o pojemności równej aktualnej wartości przepływu.

Złożoność i uwagi. Metoda Forda–Fulkersona nie precyzuje sposobu wyboru ścieżek powiększających. Dla pojemności całkowitych gwarantuje to zakończenie po skończonej liczbie augmentacji, lecz czas działania zależy od wyboru ścieżek. W skrajnym przypadku liczba iteracji może być bardzo duża. Dwa popularne uszczegółowienia tej metody to:

- Edmonds–Karp: wybiera najkrótszą ścieżkę (BFS), co daje złożoność $O(|V| \cdot |E|^2)$,
- Dinic: pracuje fazami na grafie warstwowym, osiągając $O(|V|^2 \cdot |E|)$ w ogólnym przypadku.

Uwagi implementacyjne. W implementacji praktyczne jest przechowywanie dla każdej krawędzi jej krawędzi odwrotnej (reverse), co umożliwia szybkie aktualizacje pojemności rezydualnych przy augmentacji. Wyszukiwanie ścieżki powiększającej można realizować dowolnym przeszukiwaniem grafu (DFS/BFS), zależnie od wybranej warianty metody.

2.2 Edmondsa-Karpa

Algorytm Edmondsa–Karpa jest wariantem metody Forda–Fulkersona wyznaczania maksymalnego przepływu w sieci przepływowej $G = (V, E)$ z wyróżnionym źródłem s i ujściem t oraz pojemnościami $c(u, v) \geq 0$. Kluczową cechą EK jest to, że w każdej iteracji wybiera on ścieżkę powiększającą o najmniejszej liczbie krawędzi, tzn. znajduje ją algorytmem BFS w grafie rezydualnym.

Graf rezydualny. Dla aktualnego przepływu f definiuje się pojemność rezydualną

$$c_f(u, v) = c(u, v) - f(u, v),$$

a także krawędzie wsteczne, które umożliwiają cofanie wcześniej wysłanego przepływu: jeśli $f(u, v) > 0$, to w grafie rezydualnym istnieje krawędź (v, u) o pojemności rezydualnej $c_f(v, u) = f(u, v)$. Zbiór krawędzi rezydualnych tworzy graf G_f.

Iteracja algorytmu. W każdej iteracji:

1. Uruchamiamy BFS w grafie rezydualnym G_f zaczynając od s, aby znaleźć najkrótszą (w liczbie krawędzi) ścieżkę powiększającą do t.
2. Jeśli t jest nieosiągalne, to nie istnieje już ścieżka powiększająca i obecny przepływ jest maksymalny.
3. W przeciwnym razie wyznaczamy *wąskie gardło* ścieżki P:

$$\Delta = \min_{(u,v) \in P} c_f(u, v),$$

   a następnie powiększamy przepływ o Δ na krawędziach ścieżki P. Dla krawędzi zgodnych z kierunkiem zwiększamy $f(u, v)$, natomiast dla krawędzi wstecznych zmniejszamy $f(u, v)$ (co odpowiada przesuwaniu przepływu na alternatywne trasy).

Każde takie powiększenie zwiększa wartość przepływu o Δ i jest liczone jako jedna *ścieżka powiększająca* (użyte w eksperymentach jako licznik iteracji algorytmu).

Złożoność. W EK każda iteracja znajduje ścieżkę BFS w czasie $O(|E|)$. Można wykazać, że liczba iteracji jest ograniczona przez $O(|V| \cdot |E|)$, ponieważ odległość (w sensie BFS) najkrótszej ścieżki od s do dowolnego wierzchołka w grafie rezydualnym nie maleje, a „krawędzie krytyczne" (saturujące się na ścieżkach) mogą zmieniać swój status ograniczoną liczbę razy. W konsekwencji całkowita złożoność czasowa wynosi

$$O(|V| \cdot |E|^2).$$

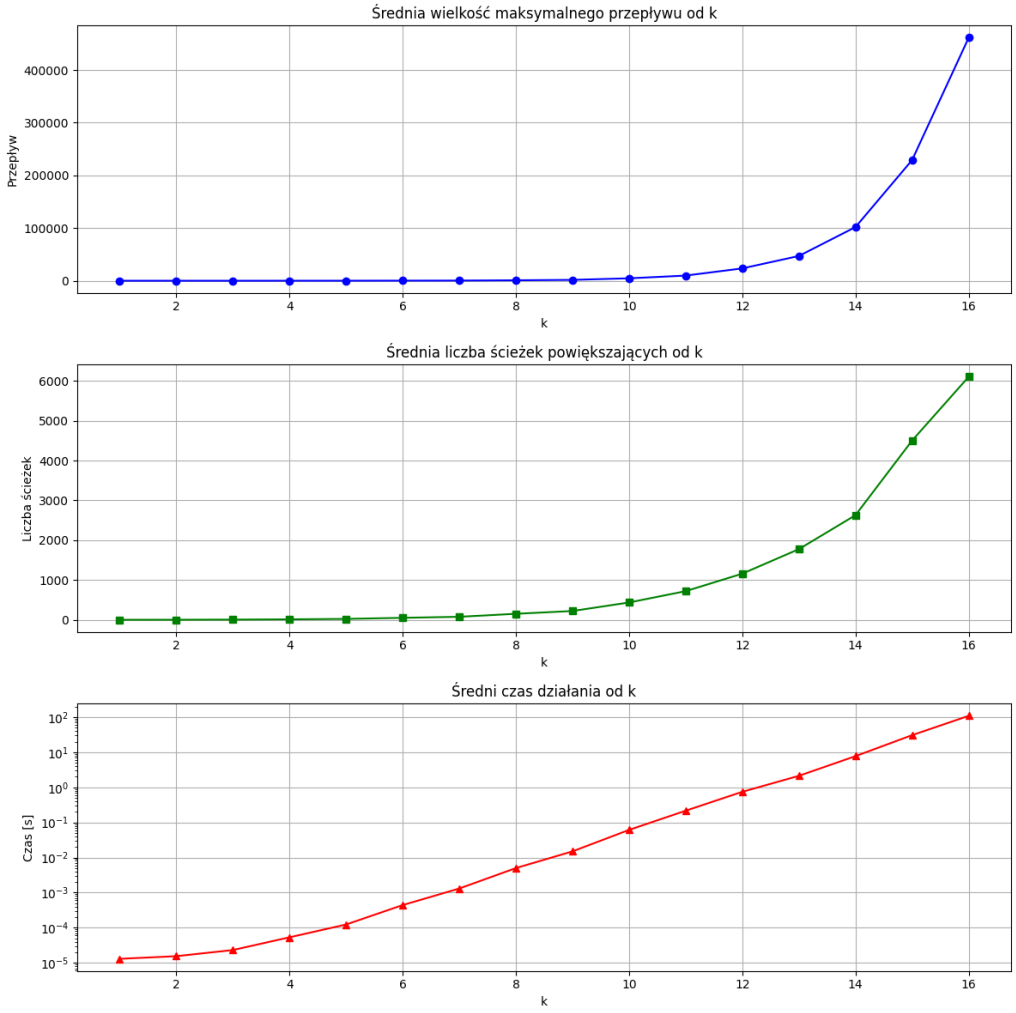
Uwagi implementacyjne. W implementacji wygodnie jest przechowywać sieć w postaci list sąsiedztwa z parami krawędzi: krawędzią *forward* o pojemności rezydualnej oraz krawędzią *reverse* umożliwiającą cofanie przepływu. BFS zapisuje poprzednika wierzchołka i indeks krawędzi, co pozwala po znalezieniu t odtworzyć ścieżkę, wyznaczyć Δ, a następnie zaktualizować pojemności rezydualne na krawędziach forward i reverse.

3 Zadanie 1

Cel Celem zadania 1 było wyznaczenie maksymalnego przepływu w sieciach będących skierowanymi hiperkostkami H_k dla kolejnych wartości k oraz czasu działania algorytmów Edmondsa–Karpa.

Metodyka Dla każdej wartości k generowano instancję hiperkostki z losowymi pojemnościami krawędzi (z wykorzystaniem stałego ziarna `seed`, aby zapewnić powtarzalność). Każdy pomiar wykonano kilkukrotnie, a następnie uśredniono czasy. Dla algorytmu własnych EK mierzono czas wykonania programu.

Wyniki Na Figure 1 przedstawiono zależność czasu działania od parametru k (w skali logarytmicznej na osi czasu). Widać, że wraz ze wzrostem rozmiaru instancji czas działania rośnie znacząco.



Rysunek 1: Zadanie 1: wszystkie wykresy.

Dodatkowa obserwacja. Wartość maksymalnego przepływu zależy od wylosowanych pojemności, dlatego w eksperymentach istotne jest użycie stałego ziarna losowości. Dla pełności na Figure 1 pokazano przykładową zależność wartości maksymalnego przepływu od k.

Podsumowanie. Dla hiperkostek H_k wraz ze wzrostem k rośnie rozmiar sieci ($|V| = 2^k$, $|E| = k \cdot 2^{k-1}$), co silnie wpływa na czas obliczeń.

4 Zadanie 2

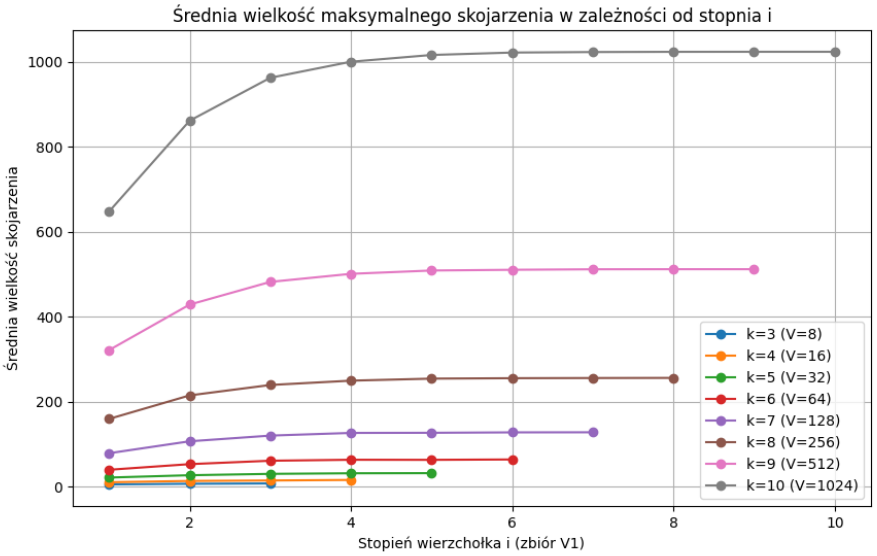
Wstęp Mamy dany `nieskierowany graf dwudzielny` zdefiniowany jako: $G_{k,i} = (V_1 \cup V2, E)$, gdzie V_1 i V_2 to rozłączne podgrafy G mające po 2^k elementów, a zbiór E, to zbiór krawędzi łączcy prawą i lewą stronę.

Cel Celem zadania 2 było znalezienie maksymalnego skojarzenia w losowym grafie dwudzielnym $G_{k,i}$ o rozmiarach $|V_1| = |V_2| = 2^k$, w którym każdy wierzchołek po lewej stronie ma dokładnie i sąsiadów po prawej stronie. Problem sprowadzono do maksymalnego przepływu przez standardową konstrukcję sieci: źródło s połączone z każdym $u \in V_1$ krawędzią o pojemności 1, krawędzie $u-v$ (dla $(u, v) \in E$) o pojemności 1 oraz krawędzie z $v \in V_2$ do ujścia t o pojemności 1. W takiej sieci wartość maksymalnego przepływu jest równa rozmiarowi maksymalnego skojarzenia.

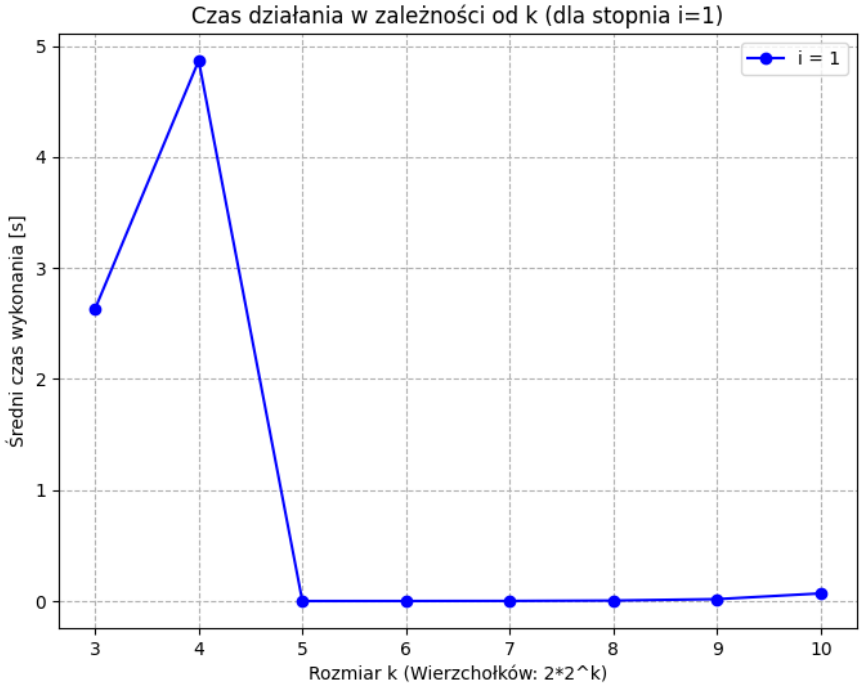
Metodyka Dla wybranych wartości k oraz stopnia i generowano losowe grafy dwudzielne, używając stałego `seed` w celu powtarzalności. Dla każdej pary parametrów wykonywano kilka uruchomień i uśredniano czasy, wykorzystując algorytm Edmondsa–Karpa.

Wyniki czasowe Na wykresach na dole przedstawiono zależność czasu działania od stopnia i (uśrednioną po rozpatrywanych wartościach k oraz powtórzeniach). Wraz ze wzrostem i graf staje się gęstszy (więcej krawędzi $V_1 \times V_2$), co zwykle zwiększa koszt obliczeń.

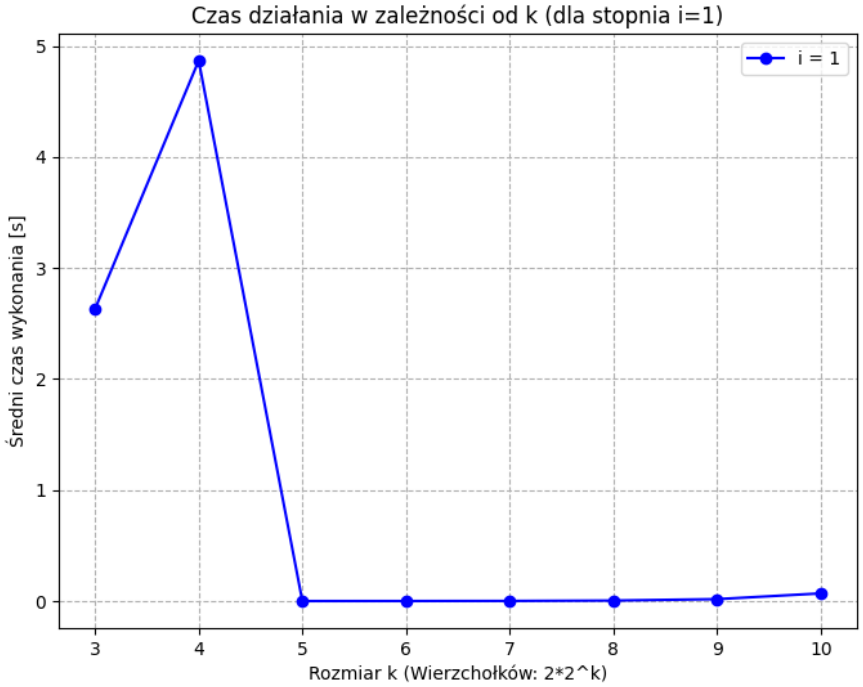
Wielkość skojarzenia Rozmiar maksymalnego skojarzenia zależy od parametrów k i i oraz konkretnej realizacji losowania. Dla małych i graf może nie zawierać doskonałego skojarzenia, natomiast przy większym i rośnie szansa na skojarzenie bliskie 2^k. Dla ilustracji na Figure 2 pokazano przykładową zależność rozmiaru skojarzenia od i.



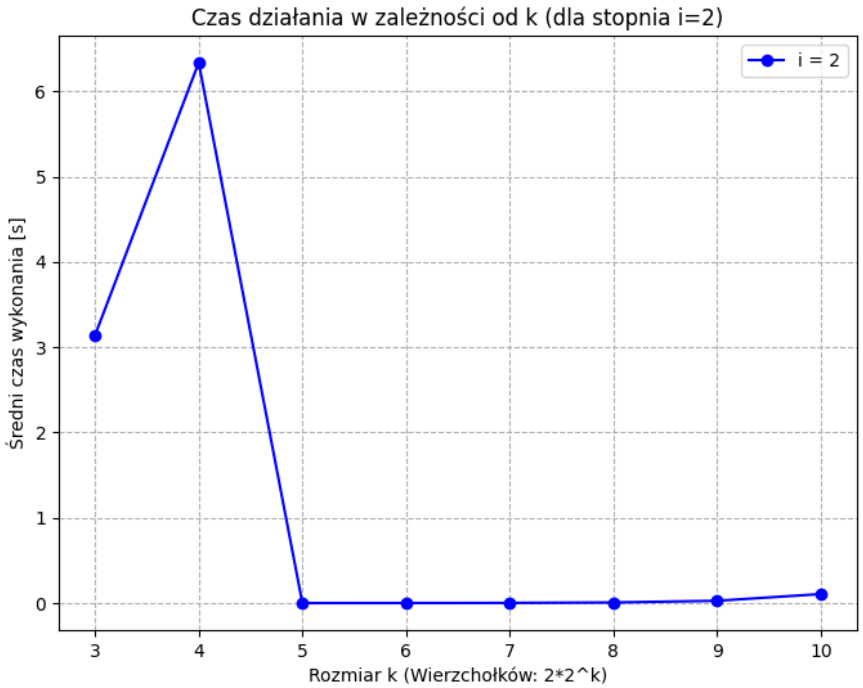
Rysunek 2: Zadanie 2: porównanie średniej wielkości maksymalnego skojarzenia w zależności od i dla każdego k. Oś wielkości w skali logarytmicznej.



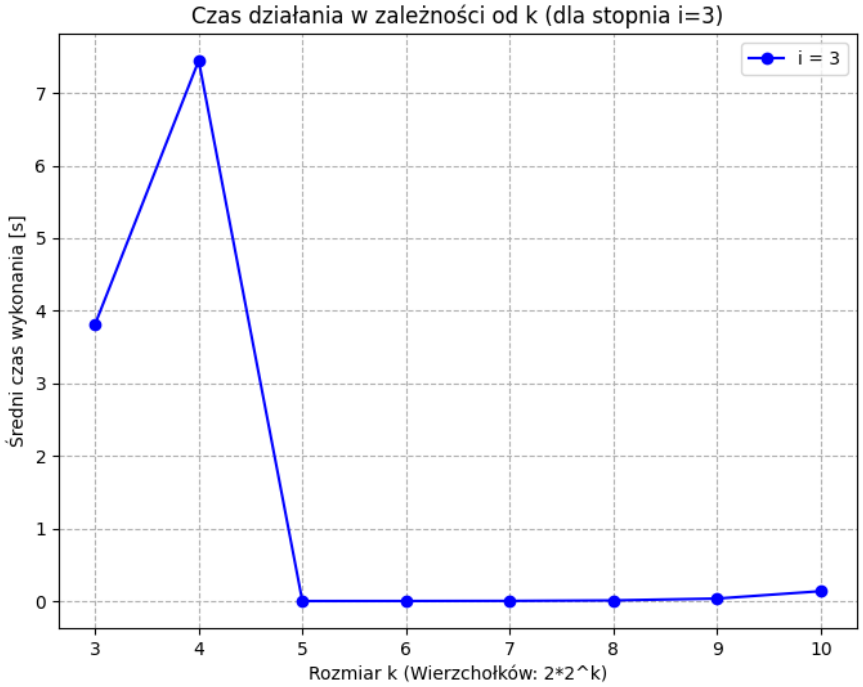
Rysunek 3: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 1$. Oś czasu w skali logarytmicznej.



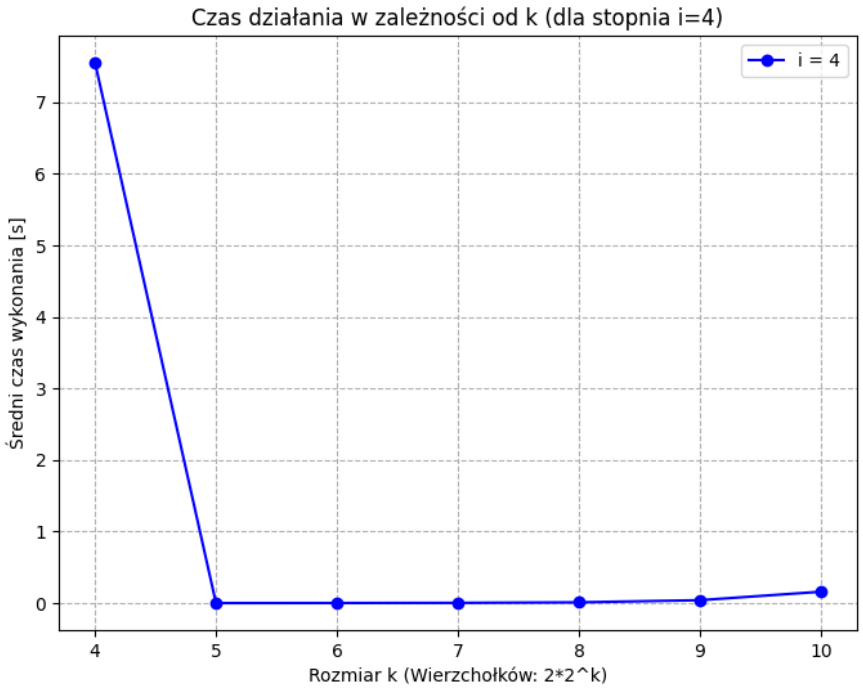
Rysunek 4: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 1$. Oś czasu w skali logarytmicznej.



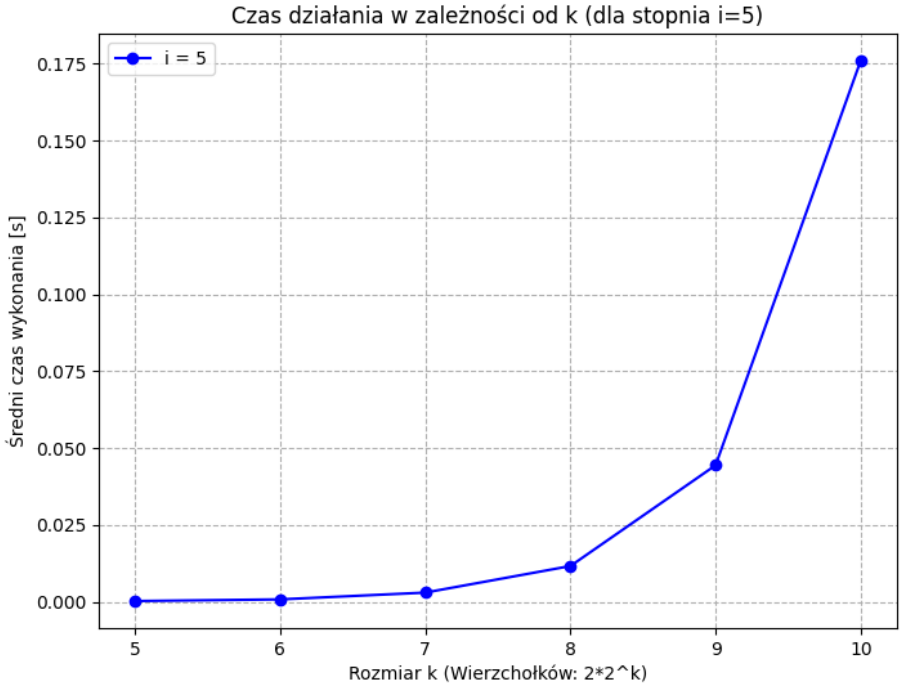
Rysunek 5: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 2$. Oś czasu w skali logarytmicznej.



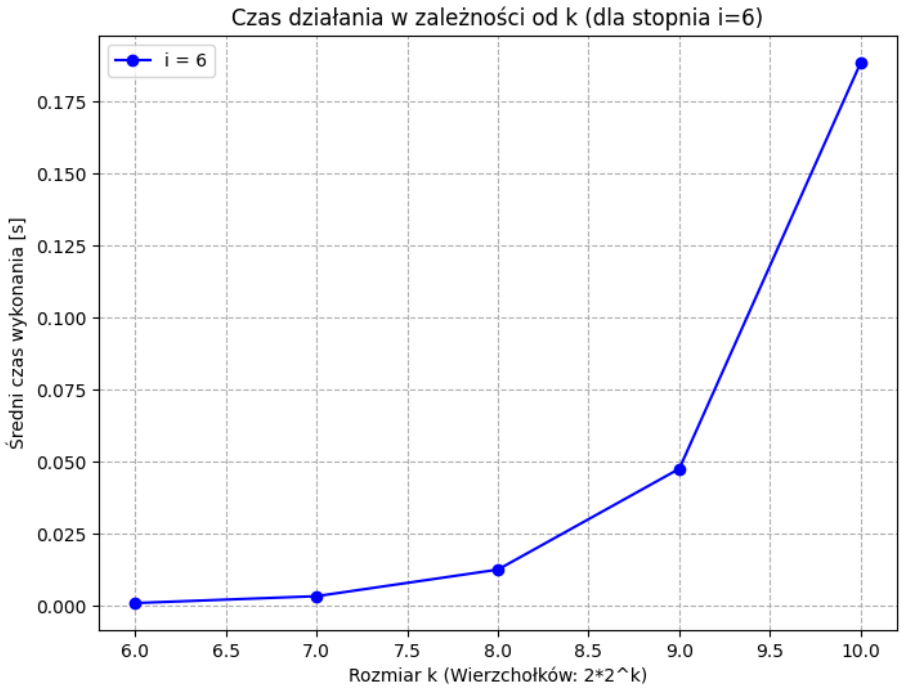
Rysunek 6: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 3$. Oś czasu w skali logarytmicznej.



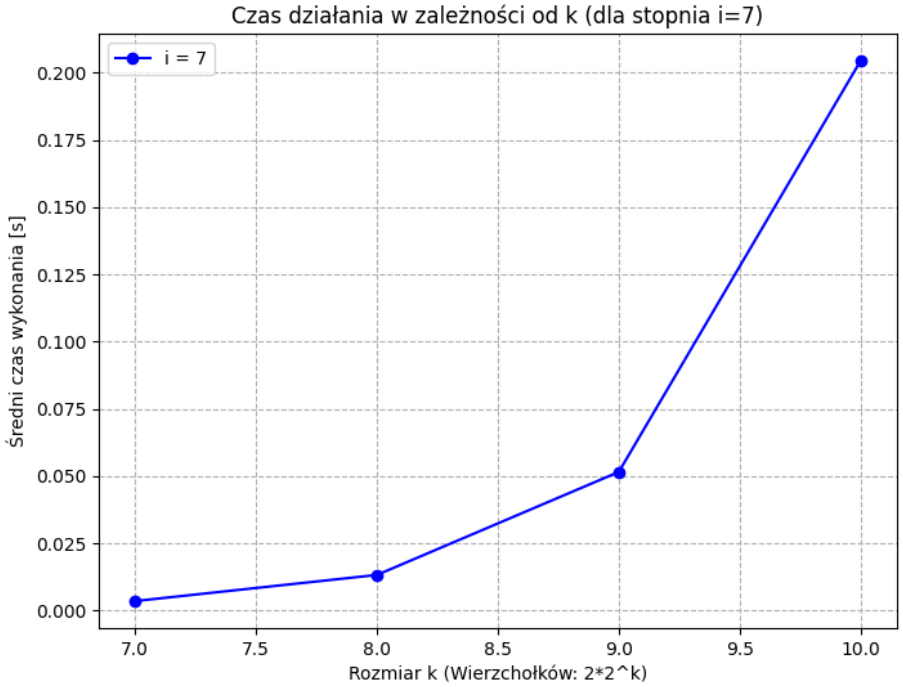
Rysunek 7: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 4$. Oś czasu w skali logarytmicznej.



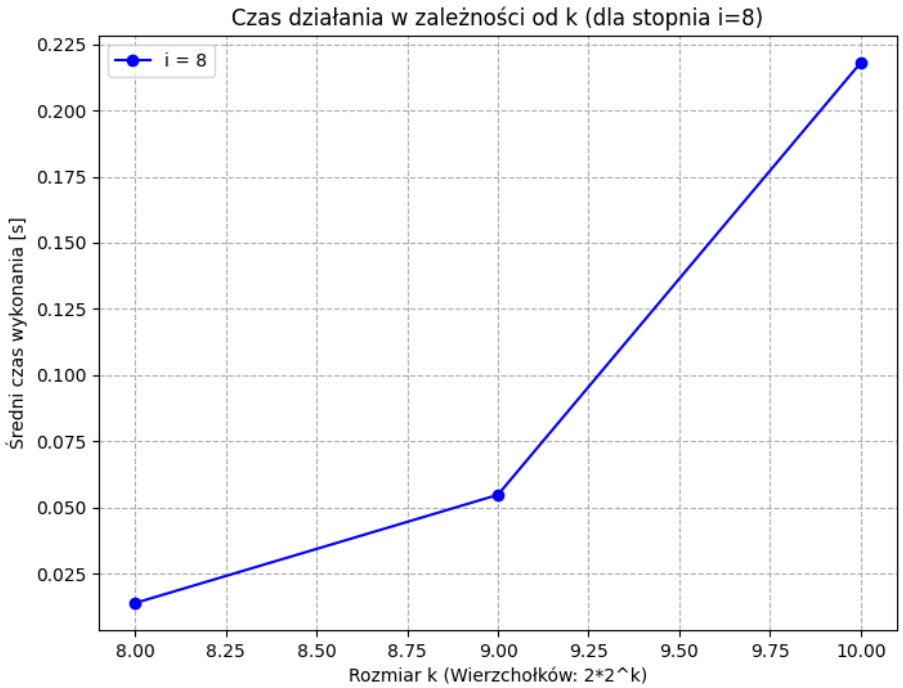
Rysunek 8: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 5$. Oś czasu w skali logarytmicznej.



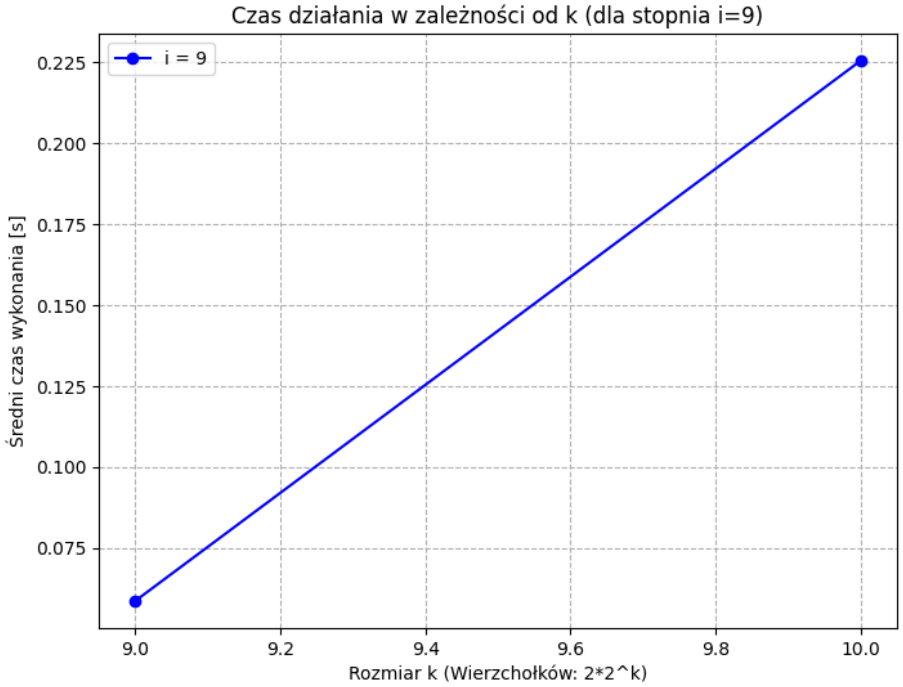
Rysunek 9: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 6$. Oś czasu w skali logarytmicznej.



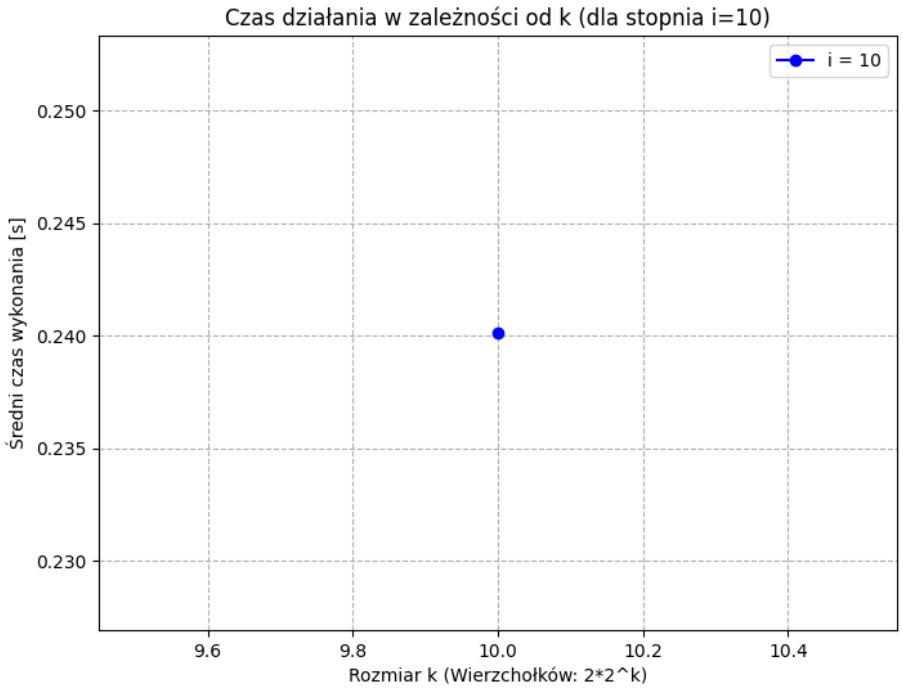
Rysunek 10: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 7$. Oś czasu w skali logarytmicznej.



Rysunek 11: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 8$. Oś czasu w skali logarytmicznej.



Rysunek 12: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 9$. Oś czasu w skali logarytmicznej.



Rysunek 13: Zadanie 2: porównanie czasu działania (średnio z kilku uruchomień) w funkcji k dla $i = 10$. Oś czasu w skali logarytmicznej.