Sprawozdanie Lista 4

Paweł Krzyszczak (nr 272379)

Styczeń 2025

1 Zadanie 1

1.1 Charakterystyka grafu

Waga Hamminga $H(x)$ dla dowolnego ciągu bitowego x to liczba jedynek w tym ciągu. Analogicznie, wartość $Z(x)$ definiujemy jako liczbę zer w ciągu x. Obie te wartości dla ciągu o długości k należą do zbioru $0, \ldots, k$.

Rozważmy skierowaną hiperkostkę H_k, gdzie $k \in 1, \ldots, 16$. Jest to graf skierowany o $|N_k| = 2^k$ wierzchołkach, których etykiety odpowiadają wszystkim możliwym ciągom binarnym długości k. Krawędzie w tym grafie łączą wierzchołki, których etykiety różnią się dokładnie na jednej pozycji, i prowadzą od wierzchołka o mniejszej wadze Hamminga do wierzchołka o większej wadze Hamminga. W każdym wierzchołku rozpoczyna się lub kończy dokładnie k łuków. W konsekwencji liczba łuków w takim grafie wynosi $|A_k| = k \cdot 2^{k-1}$.

Pojemność każdej krawędzi u_{ij} jest losowana niezależnie i jednostajnie ze zbioru $1, \ldots, 2^l$, gdzie $l = \max H(e(i)), Z(e(i)), H(e(j)),$
Tutaj $e(i)$ oznacza etykietę wierzchołka i.

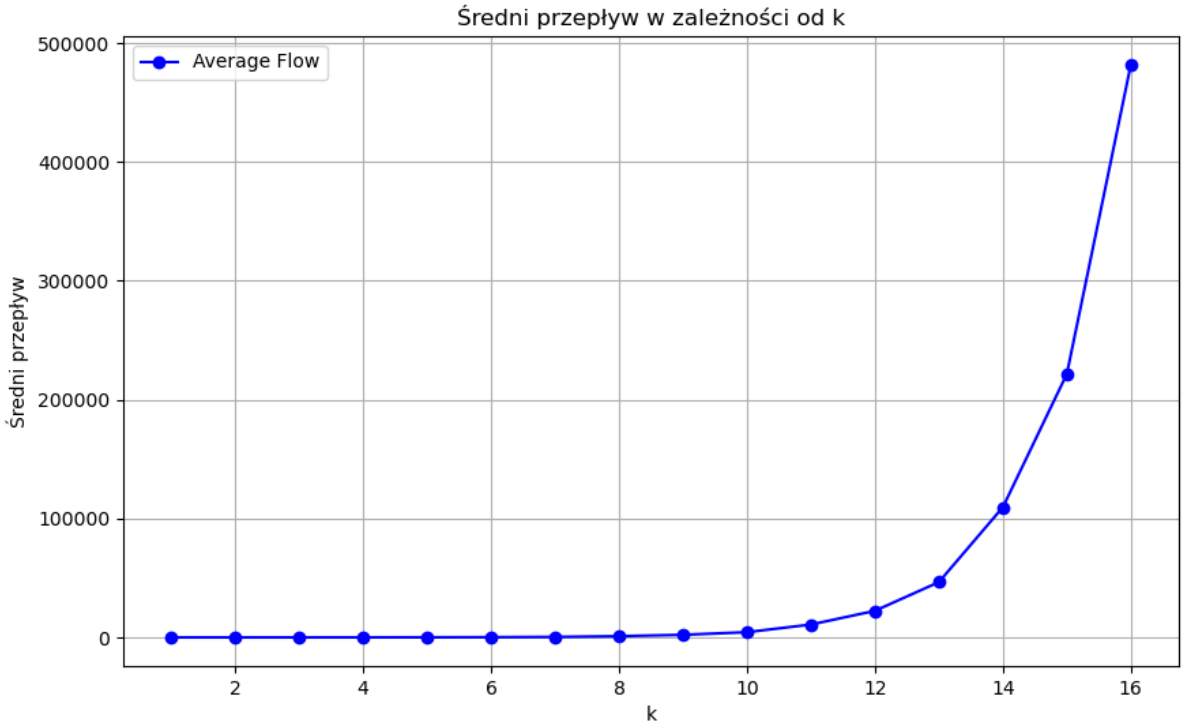
1.2 Algorytm Edmondsa-Karpa

Zakładamy, że pojemności krawędzi w grafie są nieujemne. Na podstawie tego grafu tworzymy sieć residualną. Algorytm Edmondsa-Karpa sprawdza w tej sieci, czy istnieje ścieżka powiększająca prowadząca z wierzchołka s (źródło) do wierzchołka t (ujście). Wykorzystuje w tym celu algorytm przeszukiwania wszerz (BFS). Jeśli taka ścieżka istnieje, wyznaczany jest maksymalny przepływ możliwy przez tę ścieżkę, czyli minimalna wartość pojemności krawędzi na tej ścieżce (tzw. krawędź krytyczna). Następnie aktualizowany jest całkowity maksymalny przepływ w grafie oraz wartości w sieci residualnej. Proces ten powtarza się, dopóki w sieci residualnej istnieją ścieżki powiększające.

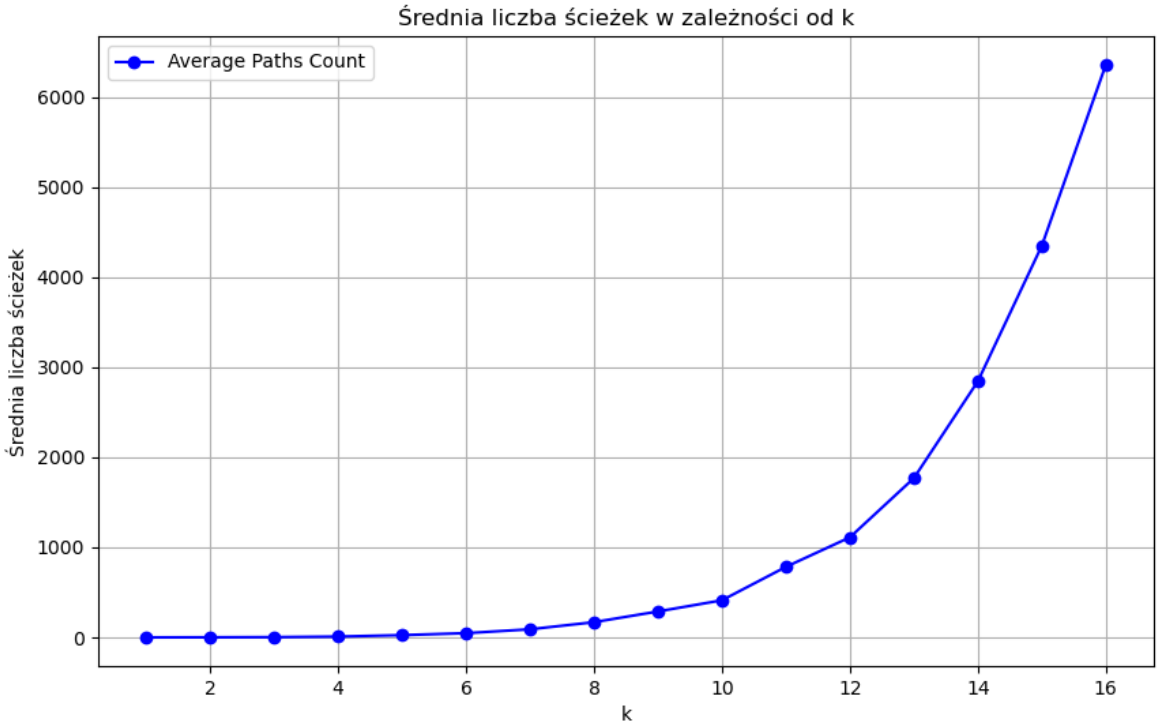
Złożoność algorytmu w najgorszym przypadku wynosi $O(|N| \cdot |A|^2)$. Algorytm BFS ma złożoność $O(|A|)$, a długości ścieżek powiększających rosną w kolejnych iteracjach. Każda krawędź w grafie może stać się krawędzią krytyczną maksymalnie $O(|N|)$ razy. Łącznie każda krawędź jest przetwarzana $O(|A|)$ razy w ramach BFS. W praktyce jednak tak wysoka złożoność jest rzadko osiągana.

1.3 Testy i ich wyniki

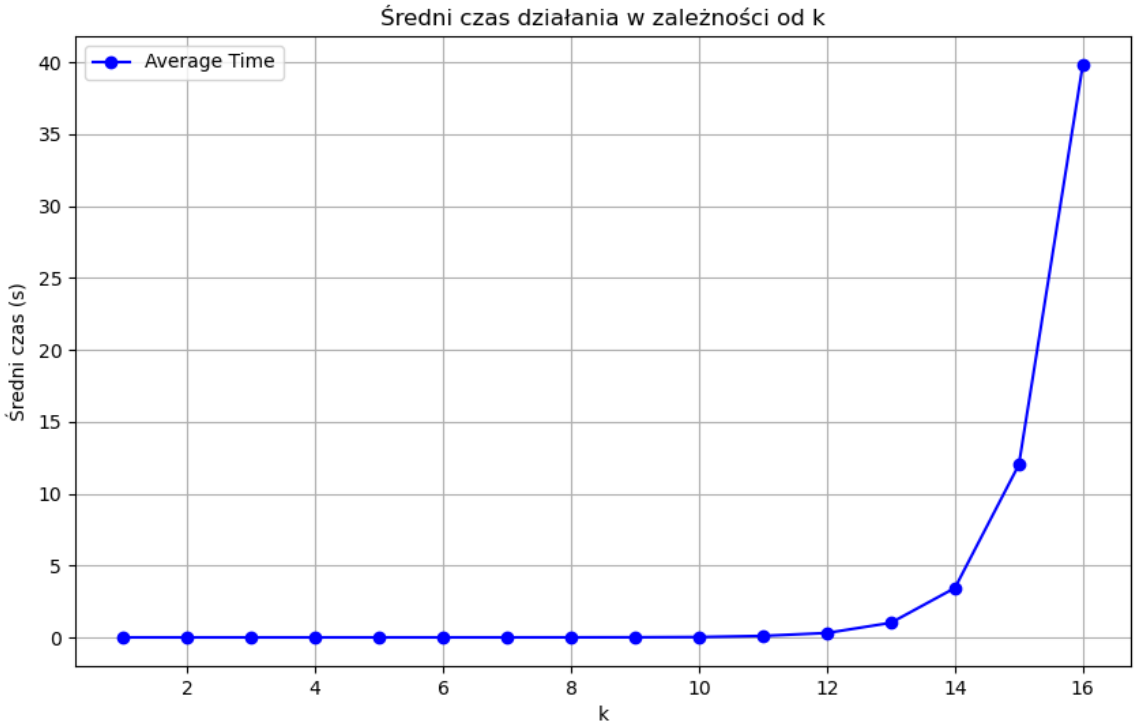
Testy algorytmu przeprowadzono na hiperkostkach H_k, gdzie $k \in \{1, \ldots, 16\}$. Dla każdej wartości k algorytm był testowany niezależnie 20 razy.



Rysunek 1: Średni przepływ w hiperkostce o wielkości k



Rysunek 2: Średnia liczba ścieżek powiększających w hiperkostce o wielkości k



Rysunek 3: Średni czas wykonywania programu dla hiperkostki o wielkości k

1.4 Wnioski

Z analizy wyników testów wynika, że wraz ze wzrostem wartości k rosną wszystkie badane wielkości. Jest to naturalna obserwacja, biorąc pod uwagę, że złożoność algorytmu wynosi $O(|N| \cdot |A|^2)$, a dla $|N| = 2^k$ i $|A| = k \cdot 2^{k-1}$ daje to złożoność względem k równą $O(2^k \cdot k^2 \cdot 2^{2k-2}) = O(4^k)$. Oznacza to, że algorytm Edmondsa-Karpa dla hiperkostki ma złożoność wykładniczą względem k, co widać wyraźnie na wykresach. Ponadto, zarówno średnia wartość przepływu, jak i średnia liczba ścieżek również rosną wykładniczo w zależności od k.

2 Zadanie 2

2.1 Charakterystyka grafu

Rozważmy $k \in \mathbb{N}$ oraz $i \in \mathbb{N}$. Przypuśćmy, że mamy graf dwudzielny, nieskierowany i losowy, który składa się z dwóch rozłącznych podzbiorów wierzchołków V_1 i V_2, z których każdy ma moc równą 2^k. Krawędzie w tym grafie są generowane w taki sposób, że każdy wierzchołek z V_1 ma i sąsiadów z V_2, przy czym sąsiedztwa są dobierane losowo, niezależnie i jednostajnie.

2.2 Algorytm

Maksymalne skojarzenie jest to taki zbiór wierzchołków w grafie, że żadne dwie krawędzie w tym zbiorze nie mają wspólnego wierzchołka.

Problem wyznaczania maksymalnego skojarzenia można sprowadzić do problemu maksymalnego przepływu. W tym celu dokonujemy kilku modyfikacji grafu dwudzielnego:

1. Krawędzie w grafie przekształcamy na krawędzie skierowane z V_1 do V_2.
2. Dodajemy wierzchołek źródła, który jest połączony krawędziami skierowanymi do wszystkich wierzchołków zbioru V_1.
3. Dodajemy wierzchołek ujścia, który jest połączony krawędziami skierowanymi do wszystkich wierzchołków zbioru V_2.
4. Ustawiamy pojemność każdej krawędzi na 1.

Na zmodyfikowanym grafie uruchamiamy algorytm do wyznaczania maksymalnego przepływu.

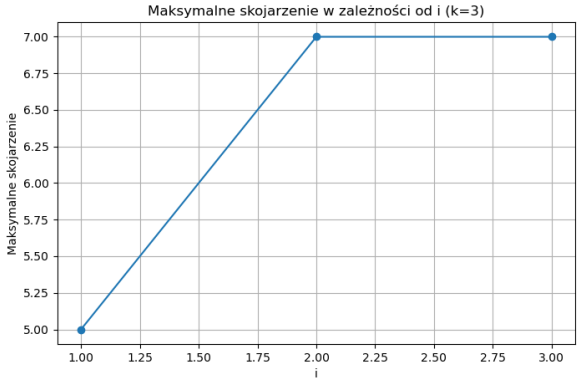
Metoda ta działa, ponieważ z każdego wierzchołka w zbiorze V_1 przepływa tylko jedna jednostka, a na każdej krawędzi łączącej wierzchołki z V_1 i V_2 maksymalny przepływ wynosi 1. Oznacza to, że każda krawędź może zostać wykorzystana tylko raz. Ponieważ z każdego wierzchołka w V_1 wypływa jedna jednostka przepływu, po wybraniu jednej krawędzi nie można już wybrać innych, co gwarantuje, że z żadnego wierzchołka z V_1 nie wybrano więcej niż jednej krawędzi. Takie same zasady obowiązują w dalszej części grafu. W efekcie, żadna krawędź w wyznaczonym zbiorze nie ma wspólnego wierzchołka, co sprawia, że zbiór ten jest maksymalnym skojarzeniem.

Aby obliczyć maksymalne skojarzenie w grafie, możemy skorzystać z wcześniej opracowanego algorytmu Edmondsa-Karpa.

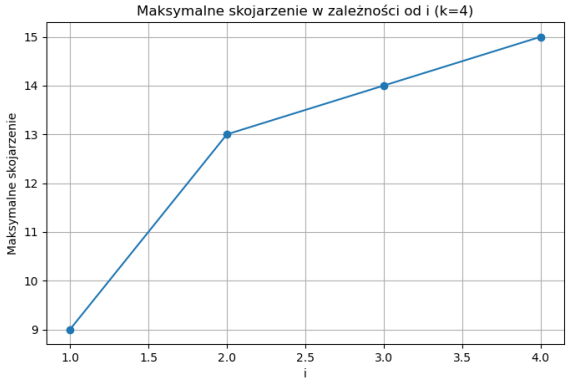
2.3 Testy i ich wyniki

Testy algorytmu zostały wykonane na grafach o $k \in \{1, \ldots, 10\}$ oraz $i \in \{1, \ldots, k\}$. Dla każdej wartości k algorytm był testowany niezależnie 20 razy.

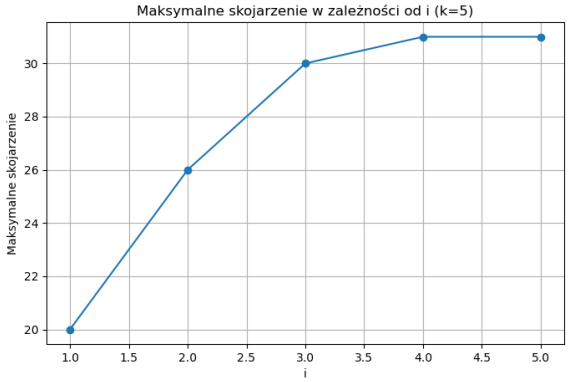
2.3.1 Wykresy maksymalnego skojarzenia



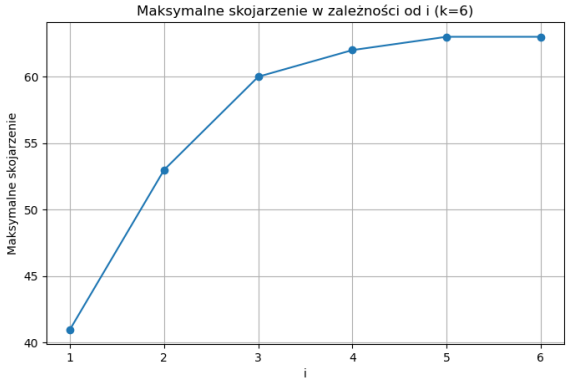
(a) Wykres dla $k = 3$.



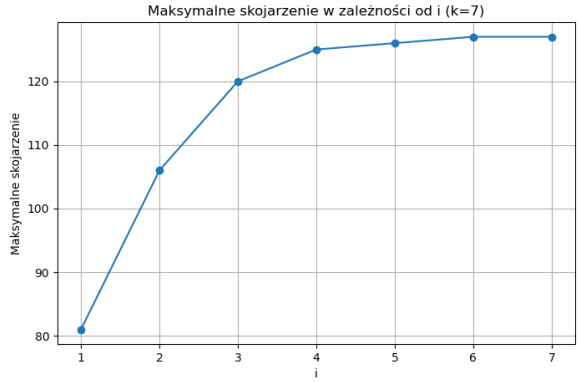
(b) Wykres dla $k = 4$.



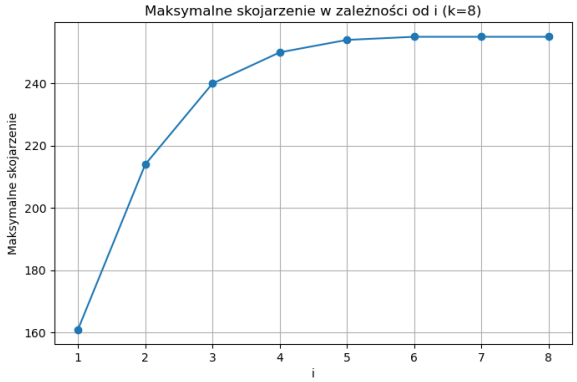
(c) Wykres dla $k = 5$.



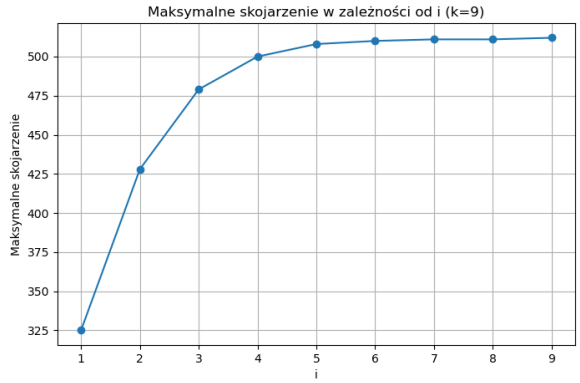
(d) Wykres dla $k = 6$.



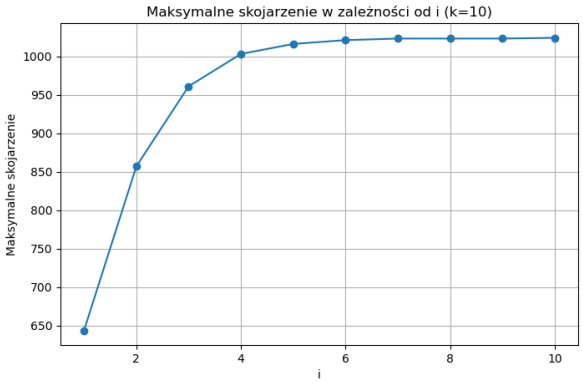
(a) Wykres dla $k = 7$.



(b) Wykres dla $k = 8$.



(c) Wykres dla $k = 9$.



(d) Wykres dla $k = 10$.

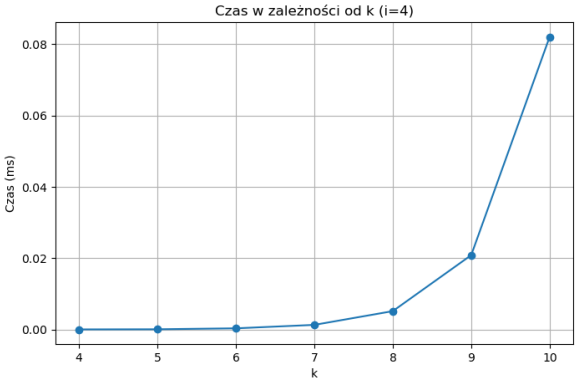
Rysunek 5: Wykresy przedstawiające wyniki dla różnych wartości k.

2.3.2 Wykresy czasu

(a) Czas dla $i = 1$.

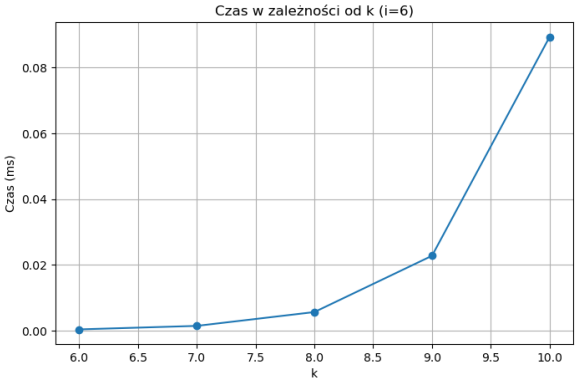
(b) Czas dla $i = 2$.

(a) Czas dla $i = 3$.

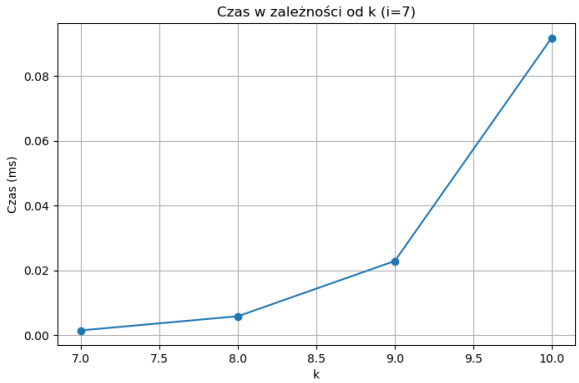


(b) Czas dla $i = 4$.

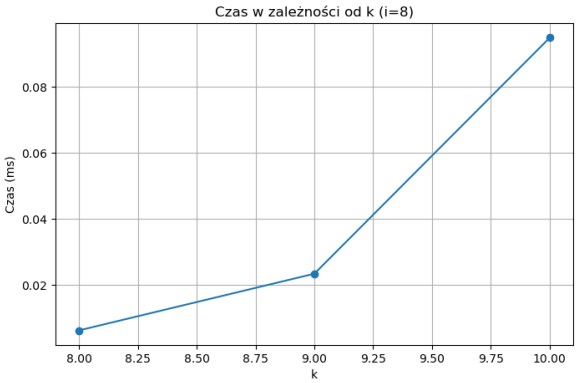
(c) Czas dla $i = 5$.



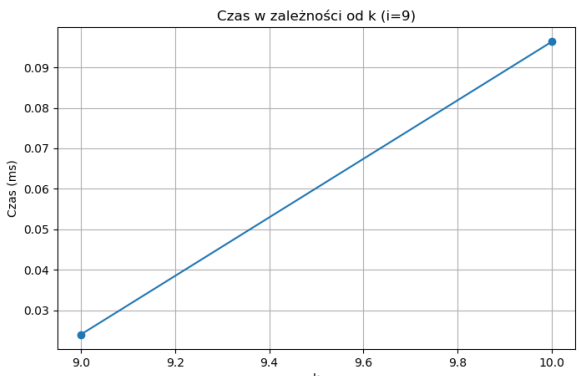
(d) Czas dla $i = 6$.

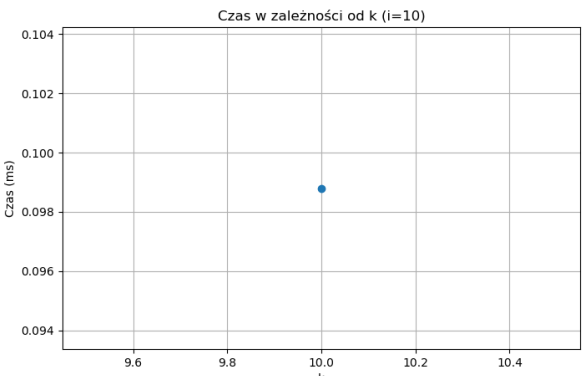


(e) Czas dla $i = 7$.



(f) Czas dla $i = 8$.





2.4 Wnioski

Z analizy wyników wynika, że czas wykonania programu rośnie wykładniczo w zależności od wartości k, a także zależy od parametru i. Złożoność algorytmu Edmondsa-Karpa, wynosząca $O(|V| \cdot |E|^2)$, wyjaśnia to zachowanie. W badanym grafie liczba wierzchołków jest równa $2 \cdot 2^k + 2$, a liczba krawędzi zależy zarówno od k, jak i od i.

Średnie maksymalne skojarzenie w badanym grafie zbliża się do wartości 2^k, co jest największym możliwym skojarzeniem, ponieważ $|V_1| = |V_2| = 2^k$, a z każdego wierzchołka wychodzi co najwyżej jedna krawędź. Już dla $i = 1$ średnie maksymalne skojarzenie przekracza połowę maksymalnej wartości, a z biegiem wzrostu i, ta wartość zbliża się do maksymalnej, osiągając ją najczęściej dla $i = k$.

Podobnie jak w pierwszym zadaniu, czas obliczeniowy wzrasta wykładniczo w zależności od k. Liczba wierzchołków w grafie wynosi $2 \cdot 2^k + 2 = O(2^k)$, a liczba krawędzi to $(2 + i) \cdot 2^k = O(2^k)$. Złożoność czasowa algorytmu wynosi więc $O(2^k \cdot 2^{k^2}) = O(4^k)$. Czas wykonania programu rośnie także w zależności od i, co jest spodziewane, ponieważ wartość ta wpływa na liczbę krawędzi w grafie, a tym samym na złożoność algorytmu.

3 Zadanie 3

3.0.1 Dane

G_{ij} - macierz wypełniona pojemnościami krawędzi grafu (0 w sytuacji gdy krawędź nie istnieje)

3.0.2 Zmienna decyzyjna

Definiujemy zmienną decyzyjną f_{ij}, która reprezentuje przepływ na krawędzi $i \to j$. Wymuszamy, by w przypadku braku takiej krawędzi w grafie wartość f była ustawiona na 0.

3.0.3 Ograniczenia

- Przepływ na każdej z krawędzi nie może być większy niż pojemność krawędzi:

$$\forall_{(i,j)\in A} f_{ij} \leq G_{ij}$$

- Dla każdego wierzchołka suma przepływu wpływającego do niego musi być równa sumie przepływów wychodzących z niego:

$$\forall_{i\in\{1,\ldots,n\}, i\neq 1, i\neq n} \sum_{j=1}^{n} f_{ij} = \sum_{j=1} f_{ji}$$

- Przepływ na każdej krawędzi musi być nieujemny:

$$f_{ij} \geq 0$$

3.0.4 Funkcja celu

Chcemy zmaksymalizować przepływ w grafie:

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}$$

3.1 Testy i ich wyniki

3.1.1 Zadanie 1

- k = 3

  Maksymalny przepływ według algorytmu Edmondsa-Karpa: 13
  Maksymalny przepływ według solvera `glpk`: 13

  Czas wykonywania algorytmu Edmondsa-Karpa: 2.3554e-05s
  Czas wykonywania dla solvera `glpk`: 8.106231689453125e-5s

Krawędź	Przepływ wg. algorytmu	Przepływ wg. solvera
$0 \to 1$	5	5
$0 \to 2$	6	6
$0 \to 4$	2	2
$1 \to 3$	1	1
$1 \to 5$	4	4
$2 \to 3$	4	4
$2 \to 6$	2	2
$3 \to 7$	5	5
$4 \to 5$	2	2
$4 \to 6$	0	0
$5 \to 7$	6	6
$6 \to 7$	2	2

Tabela 1: Maksymalny przepływ wg. algorytmu Edmondsa-Karpa oraz solvera `glpk` dla $k = 3$

- $k = 7$

  Maksymalny przepływ według algorytmu Edmondsa-Karpa: 525
  Maksymalny przepływ według solvera `glpk`: 525

  Czas wykonywania algorytmu Edmondsa-Karpa: 0.00251867s
  Czas wykonywania dla solvera `glpk`: 0.05552101135253906s

- $k = 10$

  Maksymalny przepływ według algorytmu Edmondsa-Karpa: 4620
  Maksymalny przepływ według solvera `glpk`: 4620

  Czas wykonywania algorytmu Edmondsa-Karpa: 0.0337379s
  Czas wykonywania dla solvera `glpk`: 31.606630086898804s

- $k = 11$

  Maksymalny przepływ według algorytmu Edmondsa-Karpa: 11777
  Maksymalny przepływ według solvera `glpk`: 11777

  Czas wykonywania algorytmu Edmondsa-Karpa: 0.113708s
  Czas wykonywania dla solvera `glpk`: 260.9325420856476s

- Dla wartości $k \geq 12$ działanie solvera zostawało unicestwione.

3.1.2 Zadanie 2

- $k = i = 3$

  Maksymalne skojarzenie według algorytmu Edmondsa-Karpa: 8
  Maksymalne skojarzenie według solvera `glpk`: 8

  Czas wykonywania algorytmu Edmondsa-Karpa: 3.9555e-05s
  Czas wykonywania dla solvera `glpk`: 0.0002779960632324219s

Skojarzenie wg. algorytmu	Skojarzenie wg. solvera
$1 \to 15$	$1 \to 15$
$2 \to 13$	$2 \to 13$
$3 \to 12$	$3 \to 12$
$4 \to 11$	$4 \to 11$
$5 \to 16$	$5 \to 16$
$6 \to 9$	$6 \to 9$
$7 \to 14$	$7 \to 14$
$8 \to 10$	$8 \to 10$

Tabela 2: Maksymalne skojarzenie według algorytmu Edmondsa-Karpa oraz solvera `glpk` dla $k = i = 3$

- $k = i = 7$

  Maksymalne skojarzenie według algorytmu Edmondsa-Karpa: 128
  Maksymalne skojarzenie według solvera `glpk`: 128

  Czas wykonywania algorytmu Edmondsa-Karpa: 0.00417851s
  Czas wykonywania dla solvera `glpk`: 0.5441708564758301s

- $k = i = 10$

  Maksymalne skojarzenie według algorytmu Edmondsa-Karpa: 1024
  Maksymalne skojarzenie według solvera `glpk`: 1024

  Czas wykonywania algorytmu Edmondsa-Karpa: 0.10682s
  Czas wykonywania dla solvera `glpk`: 450.0155770778656s

- Dla wartości $k = i \geq 11$ działanie solvera zostawało unicestwione.

3.2 Wnioski

Dla obu zadań solvery poprawnie wyznaczają maksymalny przepływ, choć szczegółowe rozwiązania przepływu mogą się różnić między sobą.

Jeśli chodzi o czas działania, programy korzystające z solverów były wyraźnie wolniejsze niż te, które wykorzystywały dedykowany algorytm. Przy dużych wartościach k solver mógł działać nawet kilka minut, podczas gdy algorytm Edmondsa-Karpa wykonywał to samo zadanie w ułamkach sekundy. Dla jeszcze większych wartości rozwiązanie za pomocą solvera stawało się praktycznie niemożliwe do uzyskania.

Choć solvery są wygodne w użyciu, w wielu sytuacjach zastosowanie specjalistycznego algorytmu, jeśli jest dostępny, okazuje się znacznie bardziej wydajne.