

Algorytmy i struktury danych - laboratorium

lista piąta

Paweł Rubin

Czerwiec 2019

1 Wprowadzenie

Zadania polegały na zbadaniu problemu maksymalnego przepływu oraz problemu maksymalnego skojarzenia. Poniżej znajduje się również krótki opis metody Forda-Fulkersona oraz algorytmu Edmondsa-Karpa. Na końcu tego sprawozdania badam również rozwiązanie zadań w programie glpk.

1.1 Metoda Forda-Fulkersona

Metoda Forda-Fulkersona jest stosowana do znajdowania maksymalnego przepływu w sieci przepływowej. Stanowi podstawę wielu algorytmów, między innymi algorytmu Edmondsa-Karpa czy algorytmu Dynica.

Zasadę jej działania można streścić w następujący sposób: Należy zwiększać przepływ wzdłuż dowolnej ścieżki ze źródła do ujścia, dopóki jest to możliwe.

Pseudokod

Algorithm 1 Metoda Forda-Fulkersona

```
1: while istnieje pewna ścieżka powiększająca  $p \in G_f$  do  
2:   for all  $(u, v) \in p$  do  
3:      $f(u, v) \leftarrow f(u, v) + c_f(p)$   
4:      $f(v, u) \leftarrow f(v, u) - c_f(p)$   
5:   end for  
6: end while
```

1.2 Algorytm Edmondsa-Karpa

Idea algorytmu jest identyczna z ideą metody Forda-Fulkersona, z dodatkowym warunkiem: ścieżka powiększająca, którą szukamy w każdym kroku algorytmu, musi być najkrótsza, czyli zawierać minimalną możliwą liczbę (nie wagę!) krawędzi. Taką ścieżkę znajduje się uruchamiając algorytm przeszukiwania grafu wszerek w sieci residualnej.

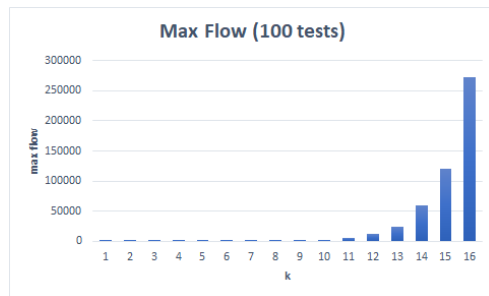
2 Zadanie pierwsze

Dla k elementowego ciągu bitowego x definiujemy wagę Hamminga $H(x)$ jako liczbę jedynek w tym ciągu (waga przyjmuje wartości od 0 do k). Niech $Z(x)$ będzie liczbą zer w ciągu x . Dla danego k rozważmy graf skierowany (k -wymiarową hiperkostkę) o 2^k wierzchołkach, których etykietami są różne binarne ciągi długości k , a krawędziami łuki między ciągami różniącymi się na dokładnie na jednej pozycji, z kierunkiem ku wierzchołkowi z większą wagą. (Łatwo

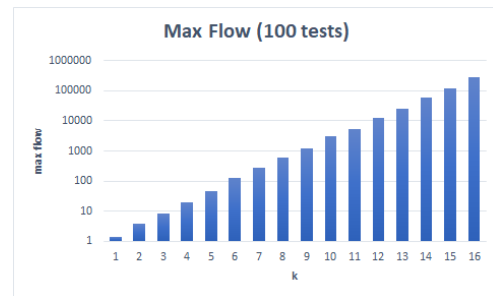
zauważyć, że każdy wierzchołek jest początkiem lub końcem dokładnie k krawędzi, a dodatkowo ciągi traktowane jako k -bitowe liczby numerują nam wierzchołki od 0 do $2^k - 1$.) Pojemność każdej krawędzi (u, v) losujemy z rozkładem jednostajnym ze zbioru $\{1, \dots, 2^l\}$, gdzie $l = \max\{H(u), Z(u), H(v), Z(v)\}$.

2.1 Testy

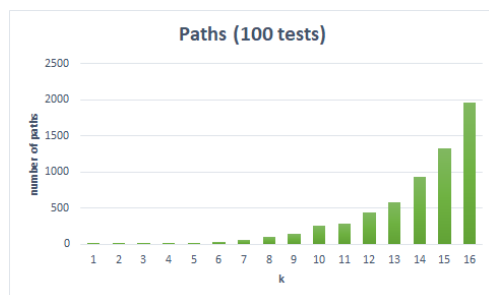
Wykonano 100 powtórzeń testów dla każdego k ze zbioru $\{1, \dots, 16\}$. Poniższe wykresy przedstawiają uśrednione wyniki.



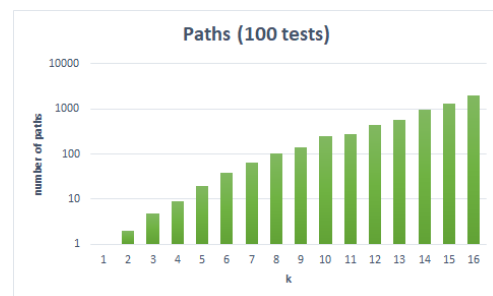
Rysunek 1: Wartość maksymalnego przepływu



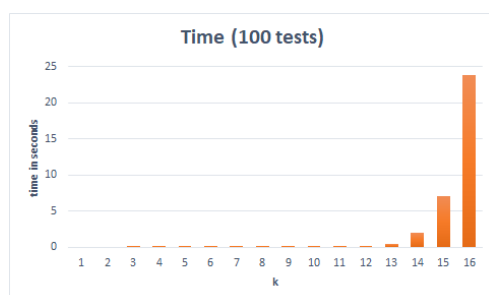
Rysunek 2: Wartość maksymalnego przepływu, skala logarytmiczna na osi y



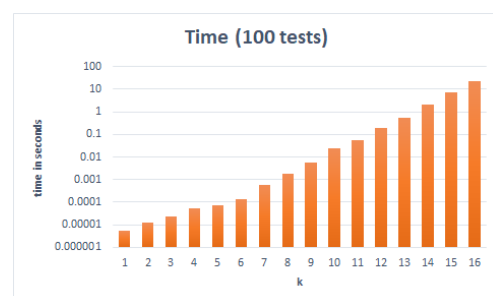
Rysunek 3: Liczba ścieżek powiększających



Rysunek 4: Liczba ścieżek powiększających, skala logarytmiczna na osi y



Rysunek 5: Czas wykonania



Rysunek 6: Czas wykonania, skala logarytmiczna na osi y

2.2 Obserwacje

Złożoność czasowa algorytmu Edmondsa-Karpa to $O(VE^2)$. W naszym problemie $V = 2^k$ a $E = \frac{k \cdot 2^k}{2}$, a więc (pomińmy stałe):

$$O(VE^2) = O(2^k * (k * 2^k)^2) = O(2^k * k^2 * 2^{2k}) = O(2^{3k} * k^2) = O(8^k * k^2)$$

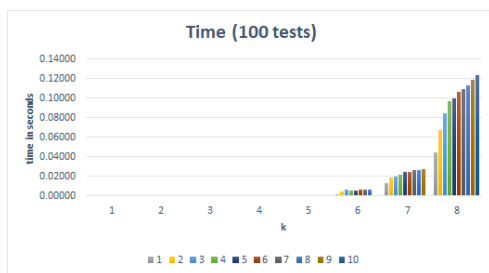
Zgadza się to z wynikowymi wykresami.

3 Zadanie drugie

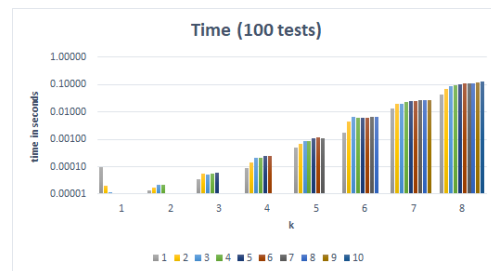
Dla danego k i i rozważamy dwudzielny graf losowy mający dwa rozłączne zbiory wierzchołków V_1 i V_2 , każdy o mocy 2^k i krawędziach losowanych jednostajnie w ten sposób, że każdy wierzchołek z V_1 ma i sąsiadów z V_2 .

3.1 Testy

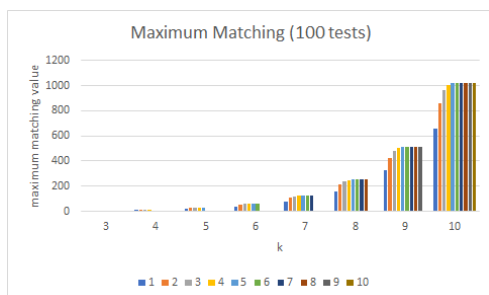
Wykonano 100 powtórzeń testów dla każdego k ze zbioru $\{1, \dots, 10\}$. Poniższe wykresy przedstawiają uśrednione wyniki.



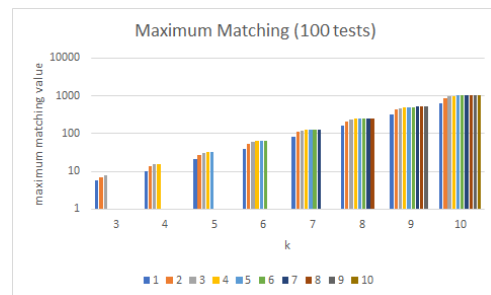
Rysunek 7: Czas wykonania



Rysunek 8: Czas wykonania, skala logarytmiczna na osi y



Rysunek 9: Wartość maksymalnego skojarzenia



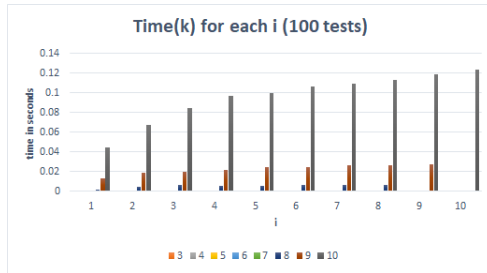
Rysunek 10: Wartość maksymalnego skojarzenia, skala logarytmiczna na osi y

3.2 Obserwacje

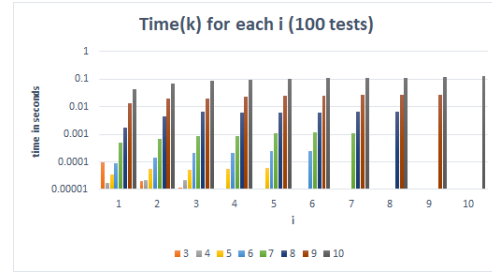
W tym przypadku $V = O(2^k)$ a $E = O(i * 2^k)$, a więc:

$$O(VE^2) = O(2^k * (i * 2^k)^2) = O(2^k * i^2 * 2^{2k}) = O(2^{3k} * i^2) = O(8^k * i^2)$$

Wobec tego, dla ustalonego k złożoność wynosi $O(i^2)$, natomiast dla ustalonego i złożoność wynosi $O(8^k)$. Obserwowane wyniki na wykresach potwierdzają tę analizę.



Rysunek 11: Czas wykonania dla k w zależności od i



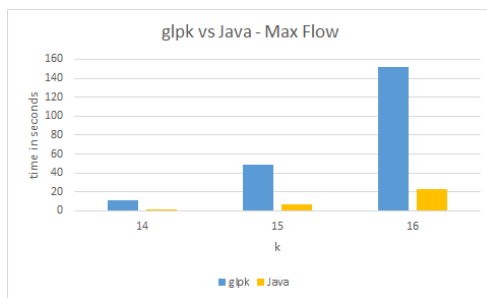
Rysunek 12: Czas wykonania dla k w zależności od i , skala logarytmiczna na osi y

4 glpk

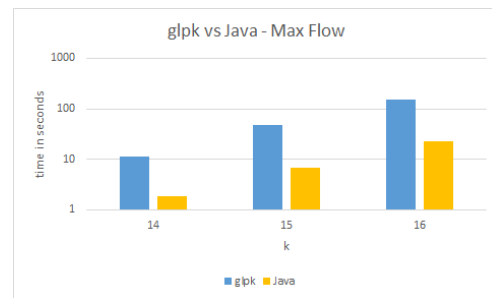
Program *glpk* daje zbliżone wyniki, aczkolwiek jego wykonanie zajmuje znacznie więcej czasu. Odpowiednio zoptymalizowana implementacja jest wielokrotnie szybsza od programu *glpk*.

4.1 Porównanie

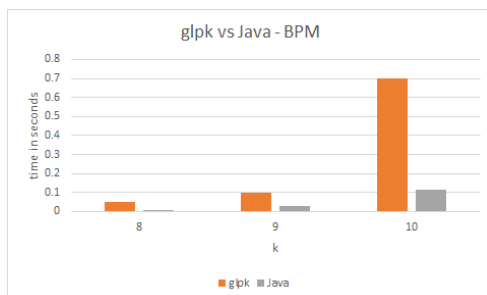
Poniższe wykresy przedstawiają porównanie czasów wykonania obliczeń, bez uwzględnienia generowania danych.



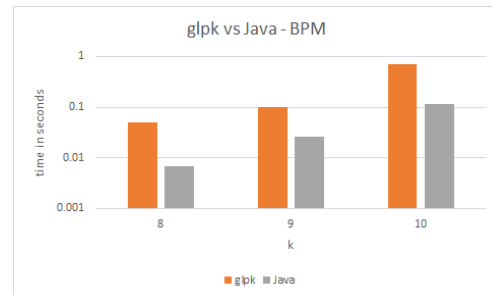
Rysunek 13: Porównanie czasu wykonania dla Max Flow



Rysunek 14: Porównanie czasu wykonania dla Max Flow, skala logarytmiczna na osi y



Rysunek 15: Porównanie czasu wykonania dla BPM



Rysunek 16: Porównanie czasu wykonania dla BPM, skala logarytmiczna na osi y

5 Podsumowanie

Problem maksymalnego skojarzenia można sprowadzić do problemu maksymalnego przepływu w sieci przepływowej o wagach równych 1, dzięki temu możemy posłużyć się jednym algoryt-

mem. Oba problemy możemy rozwiązać za pomocą programowania liniowego (program *glpk*), natomiast dzięki optymalnej implementacji, program w Javie jest znacznie szybszy.

Tabele wynikowe

k	max flow	paths	time
1	1.4	1	0.00000521
2	3.8	2	0.00001237
3	8.7	4.8	0.00002259
4	19.7	9.1	0.00005343
5	47	19.5	0.00007549
6	124.7	38.4	0.00013627
7	276.3	64.9	0.00060945
8	615	100.7	0.00172977
9	1175	139.8	0.00593085
10	3063.7	252.9	0.02385996
11	5353.1	281.4	0.05806678
12	12616.1	445.1	0.19102812
13	24676.5	582.8	0.52267239
14	59024.9	941	2.02414339
15	120403.3	1322.4	7.07741205
16	271869.3	1960	23.7635656

Tabela 1: Zadanie pierwsze - 100 testów

k	1	2	3	4	5	6	7	8	9	10
3	5.7	7.1	7.7							
4	10.1	13.7	15.3	15.9						
5	20.5	26.9	30.4	31.6	31.7					
6	40.2	53.4	60.1	62.9	63.8	63.9				
7	80.4	109.1	120.1	125.6	126.8	127.6	127.9			
8	160.3	214.6	241.1	250.3	254.4	255.3	255.8	255.7		
9	324.3	426.7	479.4	501.5	508.7	510.7	511.6	512	512	
10	654	861.6	963.1	1003	1016.7	1020	1023	1023.4	1024	1023.8

Tabela 2: Zadanie drugie: Wartości maksymalnego skojarzenia - średnia ze 100 testów

k	1	2	3	4	5	6	7	8	9	10
3	0.00001	0.00001	0.00001							
4	0.00001	0.00002	0.00002	0.00002						
5	0.00003	0.00005	0.00005	0.00006	0.00006					
6	0.00009	0.00014	0.00022	0.00021	0.00025	0.00024				
7	0.00051	0.00070	0.00084	0.00088	0.00105	0.00114	0.00110			
8	0.00171	0.00445	0.00659	0.00587	0.00594	0.00616	0.00635	0.00673		
9	0.01333	0.01901	0.02034	0.02224	0.02447	0.02459	0.02611	0.02653	0.02772	
10	0.04441	0.06773	0.08422	0.09661	0.09996	0.10637	0.10896	0.11266	0.11871	0.12393

Tabela 3: Zadanie drugie: Czas wykonania w sekundach - średnia ze 100 testów