Sprawozdanie 1 - Algorytmy Optymalizacji Dyskretnej

Michał Kallas

Październik 2024

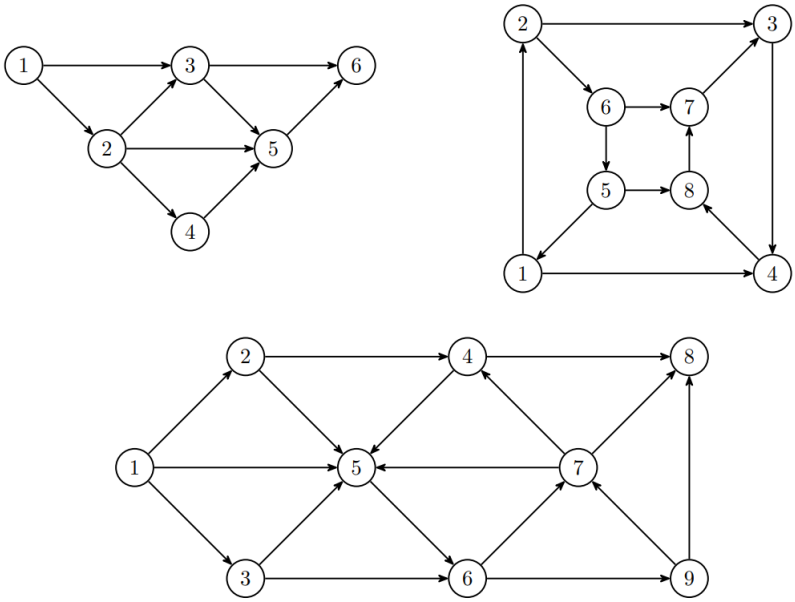
1 Zadanie 1

1.1 Opis i rozwiązanie

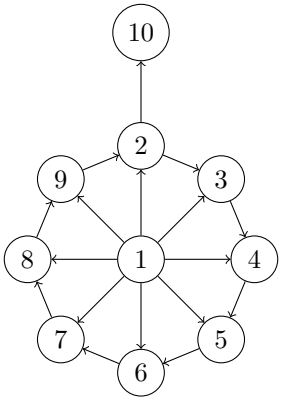
W zadaniu należało zaimplementować algorymty BFS i DFS z możliwością wyświetlania kolejności przejścia po grafie oraz drzew przeszukań.

Zaimplementowałem iteracyjne wersje tych algorytmów z kolejką i stosem. Ich złożoność to $O(|V| + |E|)$.

1.2 Grafy



Rysunek 1: Grafy testowe z polecenia, numerowane od 1 do 3 od lewej do prawej.



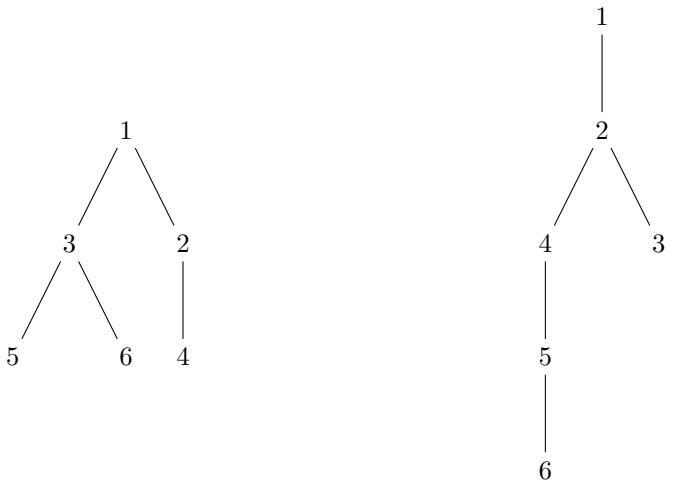
Rysunek 2: Mój graf testowy - graf nr 4.

1.3 Wyniki

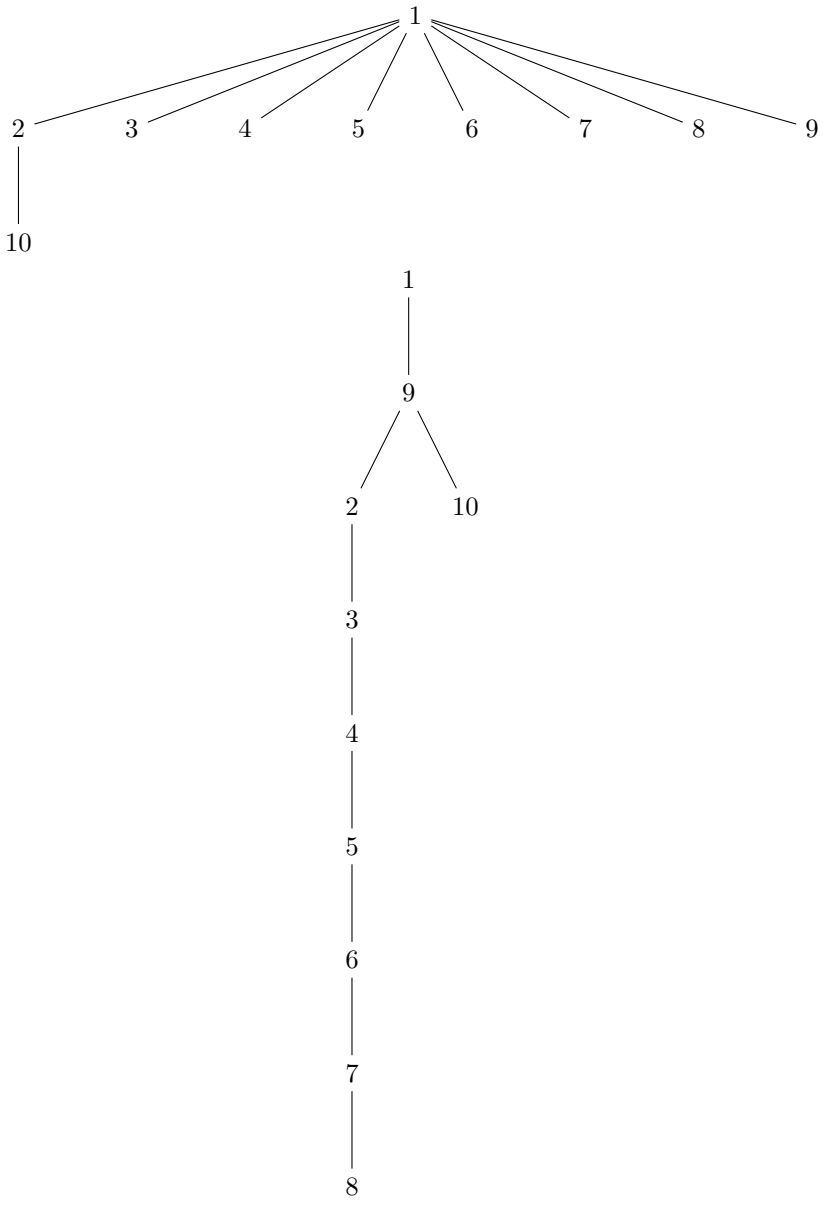
Graf	Skierowany	Kolejność BFS	Kolejność DFS
1	NIE	[1 3 2 5 6 4]	[1 2 4 5 6 3]
2	NIE	[1 2 4 5 3 6 8 7]	[1 5 6 7 8 4 3 2]
3	NIE	[1 2 3 5 4 6 7 8 9]	[1 5 7 9 8 4 2 6 3]
4	NIE	[1 2 3 4 5 6 7 8 9 10]	[1 9 2 3 4 5 6 7 8 10]
1	TAK	[1 3 2 5 6 4]	[1 2 4 5 6 3]
2	TAK	[1 2 4 3 6 8 5 7]	[1 4 8 7 3 2 6 5]
3	TAK	[1 2 3 5 4 6 8 7 9]	[1 5 6 9 8 7 4 3 2]
4	TAK	[1 2 3 4 5 6 7 8 9 10]	[1 9 2 3 4 5 6 7 8 10]

Tabela 1: Porównanie kolejności przechodzenia wierzchołków przez BFS i DFS.

1.4 Grafy



Rysunek 3: Drzewo przeszukań BFS(po lewej) oraz DFS(po prawej) dla pierwszego grafu skierowanego.



Rysunek 4: Drzewo przeszukań BFS(u góry) oraz DFS(na dole) dla mojego grafu skierowanego.

1.5 Czasy

Graf	Skierowany	$\|V\|$	$\|E\|$	Czas BFS [ns]	Czas DFS [ns]
1	NIE	6	9	5500	4200
2	NIE	8	12	2900	4400
3	NIE	9	17	3200	5900
1	TAK	6	9	1800	2200
2	TAK	8	12	2300	2600
3	TAK	9	17	2300	3200

Tabela 2: Porównanie czasów wykonania BFS i DFS dla testowych grafów.

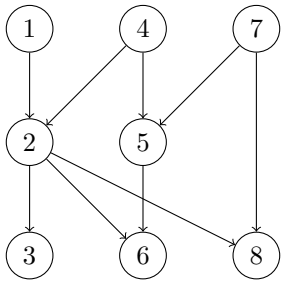
2 Zadanie 2

2.1 Opis i rozwiązanie

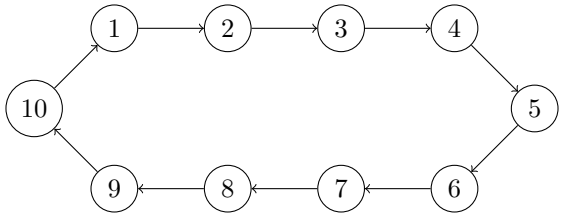
W zadaniu należało zaimplementować sortowanie topologiczne z możliwością wyświetlenia wierzchołków w posortowanej kolejności.

Zastosowałem tutaj algorytm Kahna. Jego złożoność to $O(|V| + |E|)$.

2.2 Grafy



Rysunek 5: Skierowany graf bez cyklu.



Rysunek 6: Skierowany graf z cyklem.

2.3 Wyniki

Graf	Wynik Sortowania topologicznego
1	[1 4 7 2 5 8 3 6]
2	Graf posiada cykl

Tabela 3: Porównanie wyników sortowania topologicznego dla grafu acyklicznego i cyklicznego przez algorytm Kahna.

2.4 Czasy

Plik	Acykliczny	$\|V\|$	$\|E\|$	Czas [ns]
g2a-1.txt	TAK	16	33	18700
g2a-2.txt	TAK	100	261	22500
g2a-3.txt	TAK	1600	4641	362900
g2a-4.txt	TAK	10000	29601	1745499
g2a-5.txt	TAK	160000	478401	30075586
g2a-6.txt	TAK	1000000	2996001	215613800
g2b-1.txt	NIE	16	34	6000
g2b-2.txt	NIE	100	262	27800
g2b-3.txt	NIE	1600	4642	240600
g2b-4.txt	NIE	10000	29602	1352600
g2b-5.txt	NIE	160000	478402	21810148
g2b-6.txt	NIE	1000000	2996002	164590632

Tabela 4: Porównanie czasów wykonania sortowania topologicznego algorytmem Kahna dla danych z plików testowych.

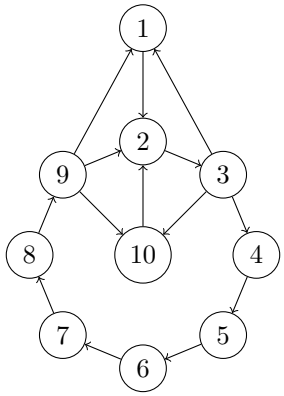
3 Zadanie 3

3.1 Opis i rozwiązanie

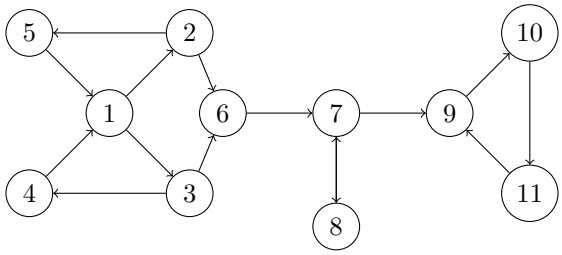
W zadaniu należało zaimplementować algorytm zwracający rozkład grafu na silnie spójne składowe. Powinien także wypisywać ich ilość oraz ilość wierzchołków w każdej z nich.

Skorzystałem tutaj z algorytmu Tarjana o złożoności $O(|V| + |E|)$.

3.2 Grafy



Rysunek 7: Graf silnie spójny.



Rysunek 8: Graf z 4 silnie spójnymi składowymi.

3.3 Wyniki

Graf	Ilość silnie spójnych składowych	Silnie spójne składowe
1	1	[10 9 8 7 6 5 4 3 2 1]
2	4	[11 10 9], [8 7], [6], [4 3 5 2 1]

Tabela 5: Porównanie podziału na silnie spójne składowe przez algorytm Tarjana.

3.4 Czasy

Plik	$\|V\|$	$\|E\|$	Czas [ns]
g3-1.txt	16	39	11275
g3-2.txt	107	185	31808
g3-3.txt	1008	1609	202491
g3-4.txt	10009	15943	3995289
g3-5.txt	100010	159679	20828123
g3-6.txt	1000011	1598897	230059301

Tabela 6: Porównanie czasów znalezienia silnie spójnych składowych algorytmem Tarjana dla danych z plików testowych.

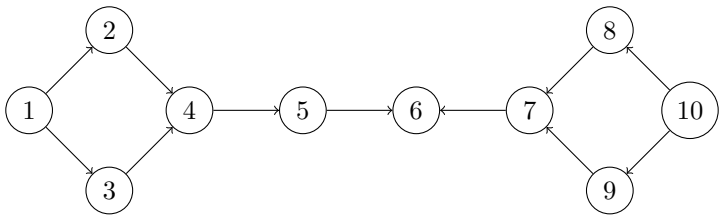
4 Zadanie 4

4.1 Opis i rozwiązanie

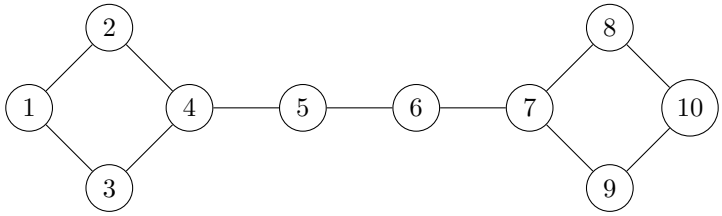
W zadaniu należało zaimplementować algorytm sprawdzający czy graf jest dwudzielny. Powinien wypisywać rozbicie grafu na 2 zbiory wierzchołków, jeśli tak jest.

Skorzystałem tutaj ze zmodyfikowanej wersji BFSa, która koloruje wierzchołki na dwa kolory. Jeśli graf można pokolorować w taki sposób, żeby sąsiednie wierzchołki nie miały tego samego koloru, to znaczy, że graf jest dwudzielny. Złożoność tego algorytmu to $O(|V| + |E|)$.

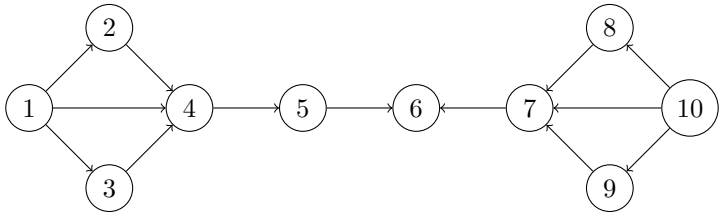
4.2 Grafy



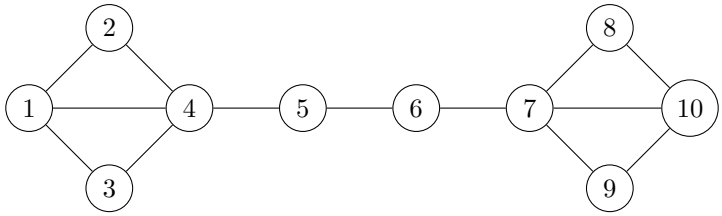
Rysunek 9: Dwudzielny graf skierowany.



Rysunek 10: Dwudzielny graf nieskierowany.



Rysunek 11: Niedwudzielny graf skierowany.



Rysunek 12: Niedwudzielny graf nieskierowany.

4.3 Wyniki

Graf	Skierowany	2 zbiory wierzchołków
1	TAK	[1 4 6 8 9], [2 3 5 7 10]
2	NIE	[1 4 6 8 9], [2 3 5 7 10]
3	TAK	Graf nie jest dwudzielny
4	NIE	Graf nie jest dwudzielny

Tabela 7: Sprawdzenie dwudzielności grafu i podział na zbiory wierzchołków za pomocą BFS z kolorowaniem.

4.4 Czasy

Plik	Skierowany	Dwudzielny	$\|V\|$	$\|E\|$	Czas [ns]
d4a-1.txt	TAK	TAK	16	24	13647
d4a-2.txt	TAK	TAK	100	180	66214
d4a-3.txt	TAK	TAK	1600	3120	1025125
d4a-4.txt	TAK	TAK	10000	19800	6286833
d4a-5.txt	TAK	TAK	160000	319200	101964833
d4a-6.txt	TAK	TAK	1000000	1998000	670709530
d4b-1.txt	TAK	NIE	16	25	12974
d4b-2.txt	TAK	NIE	100	181	59872
d4b-3.txt	TAK	NIE	1600	3121	898077
d4b-4.txt	TAK	NIE	10000	19801	5615172
d4b-5.txt	TAK	NIE	160000	319201	90784655
d4b-6.txt	TAK	NIE	1000000	1998001	580847940
u4a-1.txt	NIE	TAK	15	22	7592
u4a-2.txt	NIE	TAK	127	190	32867
u4a-3.txt	NIE	TAK	1023	1534	241987
u4a-4.txt	NIE	TAK	16383	24574	3956059
u4a-5.txt	NIE	TAK	131071	196606	31664039
u4a-6.txt	NIE	TAK	1048575	1572862	260706055
u4b-1.txt	NIE	NIE	15	22	6535
u4b-2.txt	NIE	NIE	127	190	25275
u4b-3.txt	NIE	NIE	1023	1534	182499
u4b-4.txt	NIE	NIE	16383	24574	3039818
u4b-5.txt	NIE	NIE	131071	196606	24284041
u4b-6.txt	NIE	NIE	1048575	1572862	200461039

Tabela 8: Porównanie czasów sprawdzania dwudzielności grafu za pomocą BFS z kolorowaniem dla danych z plików testowych.