AOD - sprawozdanie nr 1

Wiktor Bachta

Październik 2024

1 DFS o BFS

1.1 Algorytm

Zastosowałem standardowe algorytmy DFS i BFS. Nie skorzystałem z wersji rekurencyjnej DFS. Zastosowałem algorytm ze stosem dla DFS i algorytm z kolejką dla BFS. W alogorytmie przy wierzchołku przechowuję także jego rodzica (predecesor), aby można było odtworzyć drzewo przeszukiwania. Algorytmy mają złożoność liniową - $O(V + E)$.

1.2 Grafy

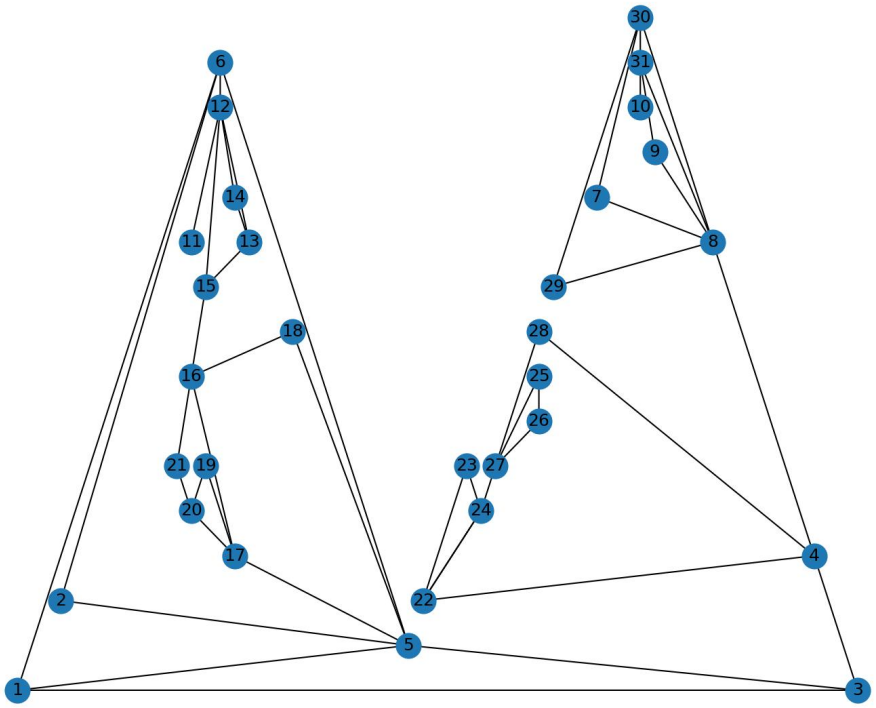

Figure 1: Graf nieskierowany

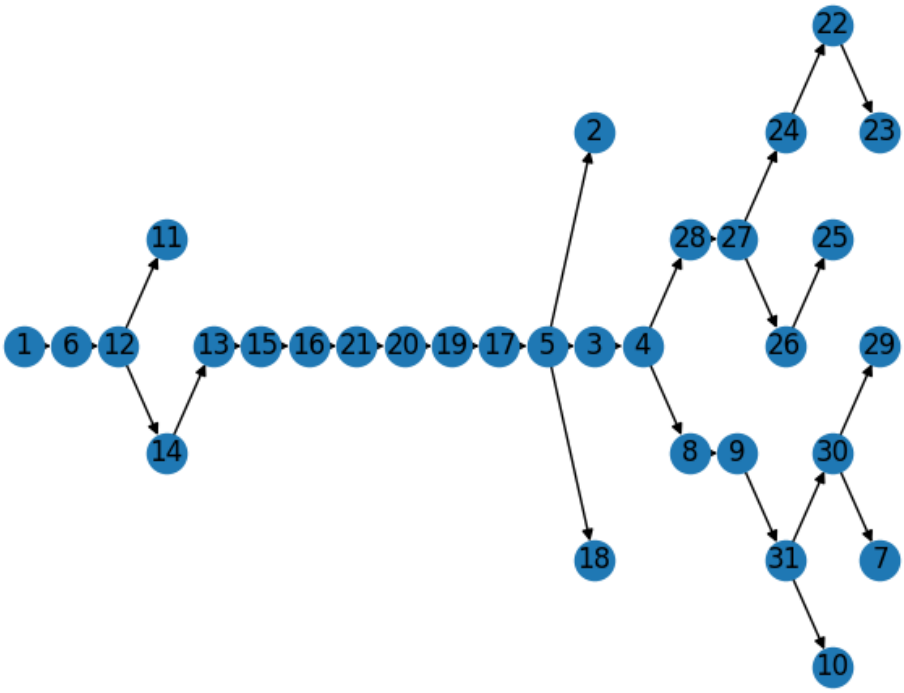

Figure 2: Drzewo DFS

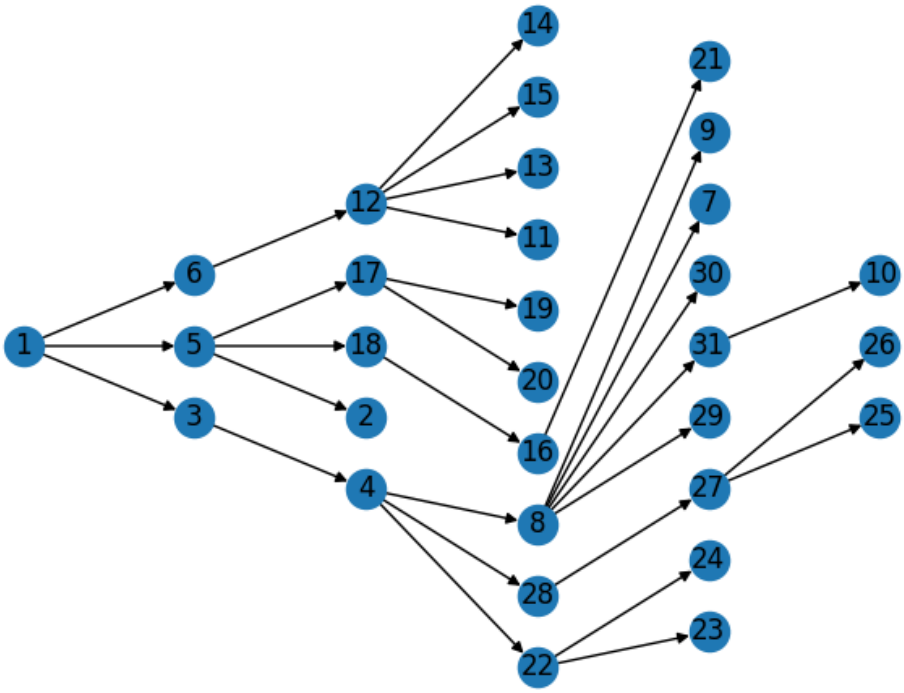

Figure 3: Drzewo BFS

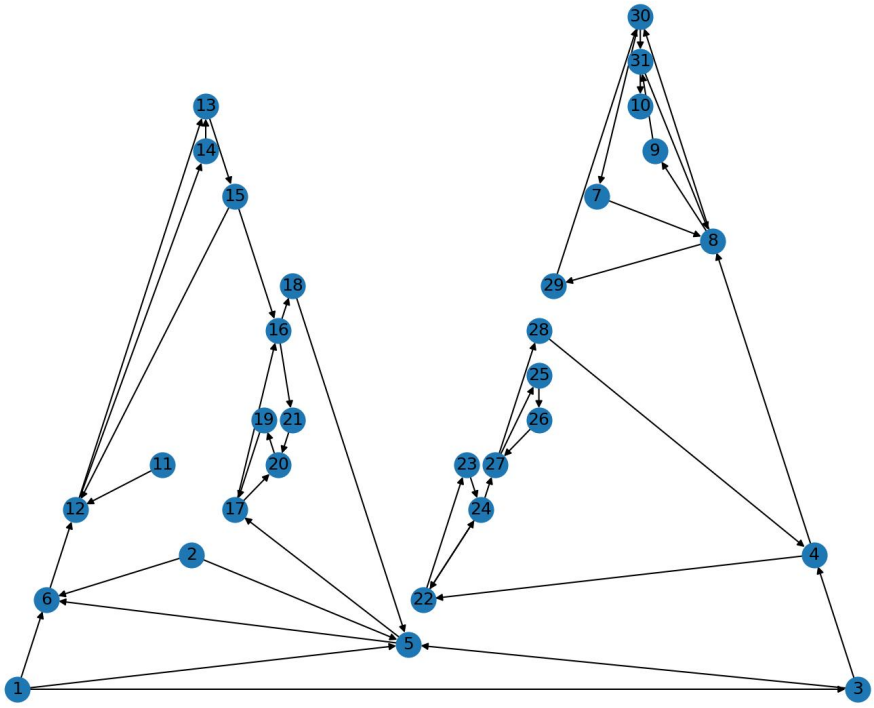

Figure 4: Graf skierowany

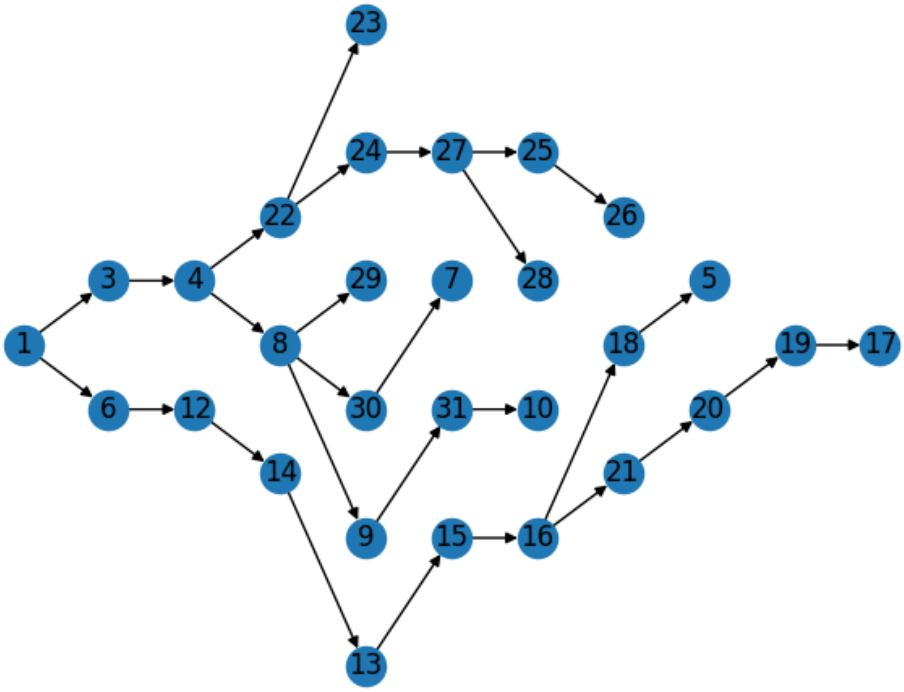

Figure 5: Drzewo DFS

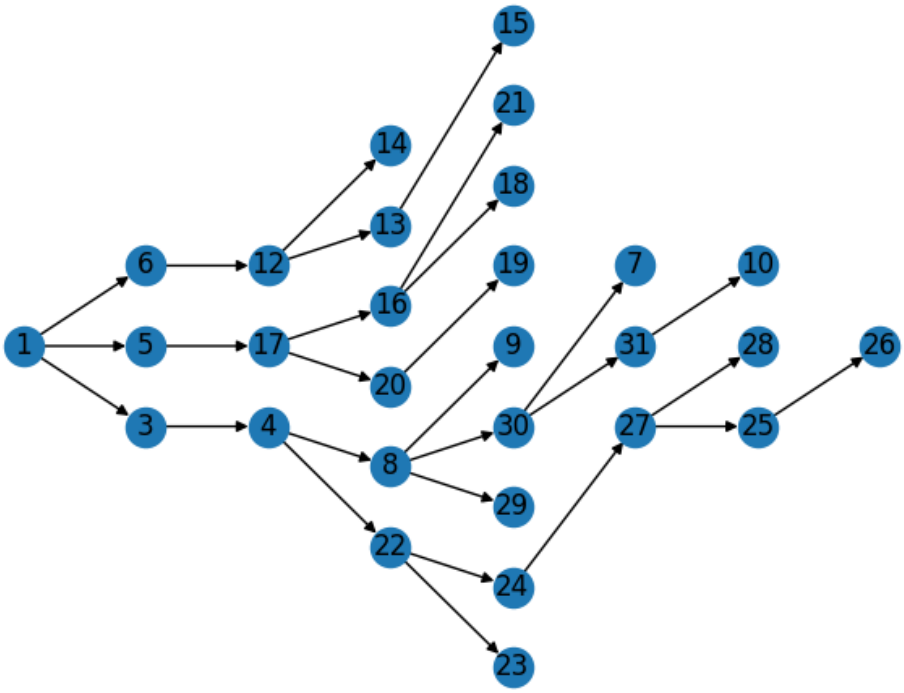

Figure 6: Drzewo BFS

1.3 Wyniki

Table 1: DFS

nr grafu	wynik
1	1-> 6-> 12-> 14-> 13-> 15-> 16-> 21-> 20-> 19-> 17-> 5-> 18-> 3-> 4-> 8-> 9-> 31-> 10-> 30-> 7-> 29-> 28-> 27-> 26-> 25-> 24-> 22-> 23-> 2-> 11
2	1-> 6-> 12-> 14-> 13-> 15-> 16-> 21-> 20-> 19-> 17-> 18-> 5-> 3-> 4-> 8-> 9-> 31-> 10-> 30-> 7-> 29-> 22-> 24-> 27-> 28-> 25-> 26-> 23

Table 2: BFS

nr grafu	wynik
1	1-> 3-> 5-> 6-> 4-> 2-> 18-> 17-> 12-> 22-> 28-> 8-> 16-> 20-> 19-> 11-> 13-> 15-> 14-> 23-> 24-> 27-> 29-> 31-> 30-> 7-> 9-> 21-> 25-> 26-> 10
2	1-> 3-> 5-> 6-> 4-> 17-> 12-> 22-> 8-> 20-> 16-> 13-> 14-> 23-> 24-> 29-> 30-> 9-> 19-> 18-> 21-> 15-> 27-> 31-> 7-> 25-> 28-> 10-> 26

1.4 Czasy

Table 3: DFS

nazwa	skierowany	|V| + |E|	czas [μs]
g3-1.txt	skierowany	55	8759
g3-2.txt	skierowany	292	12228
g3-3.txt	skierowany	2617	575506
g3-4.txt	skierowany	25952	662466
g3-5.txt	skierowany	259689	1650979
g3-6.txt	skierowany	2598908	13701511

Table 4: BFS

nazwa	skierowany	|V| + |E|	czas [μs]
g3-1.txt	skierowany	55	1793
g3-2.txt	skierowany	292	2286
g3-3.txt	skierowany	2617	12561
g3-4.txt	skierowany	25952	115375
g3-5.txt	skierowany	259689	1579917
g3-6.txt	skierowany	2598908	17243703

2 Sortowanie topologiczne

2.1 Algorytm

Zastosowałem wersję algorytmy Kahna.

- Dla każdego wierzchołka przechwuję liczbę wchodzących krawędzi c.
- Wierzchołki o $c = 0$ wrzucam na stos.
- Przy usuwaniu wierzchołka ze stosu dekremetuję c jego sukcesorów, i każdy sukcesor o $c = 0$ wrzucam na stos.
- Kolejność wierzchołków zrzucanych ze stosu to porządek topologiczny.
- Jeżeli po opróżnieniu stosu istnieje wierzchołek o $c \neq 0$, to graf posiada skierowany cykl.

Algorytm ma złożoność liniową - $O(V + E)$.

2.2 Grafy

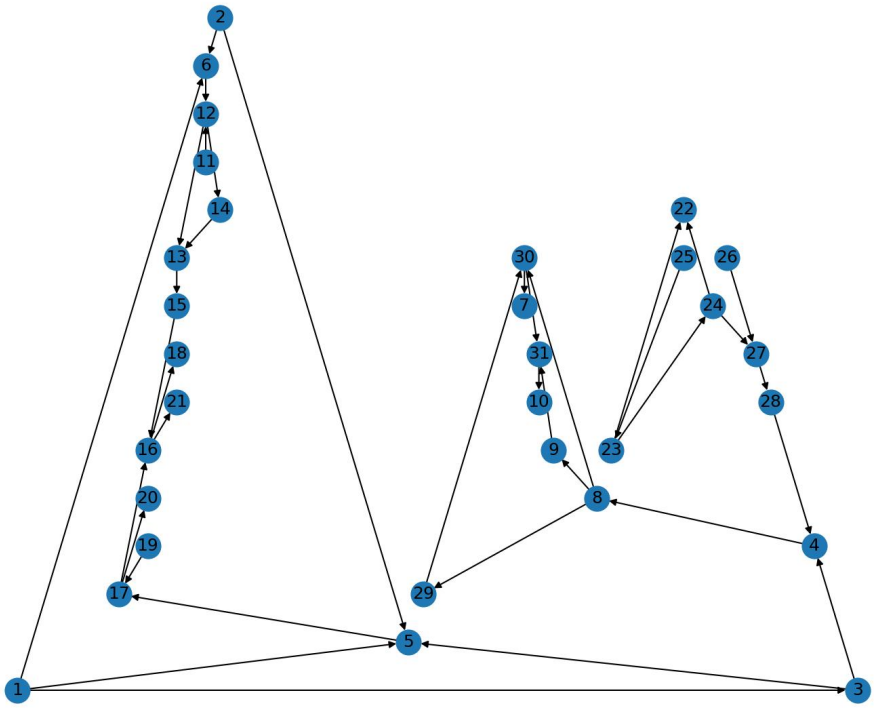

Figure 7: Graf bez cyklu skierowanego

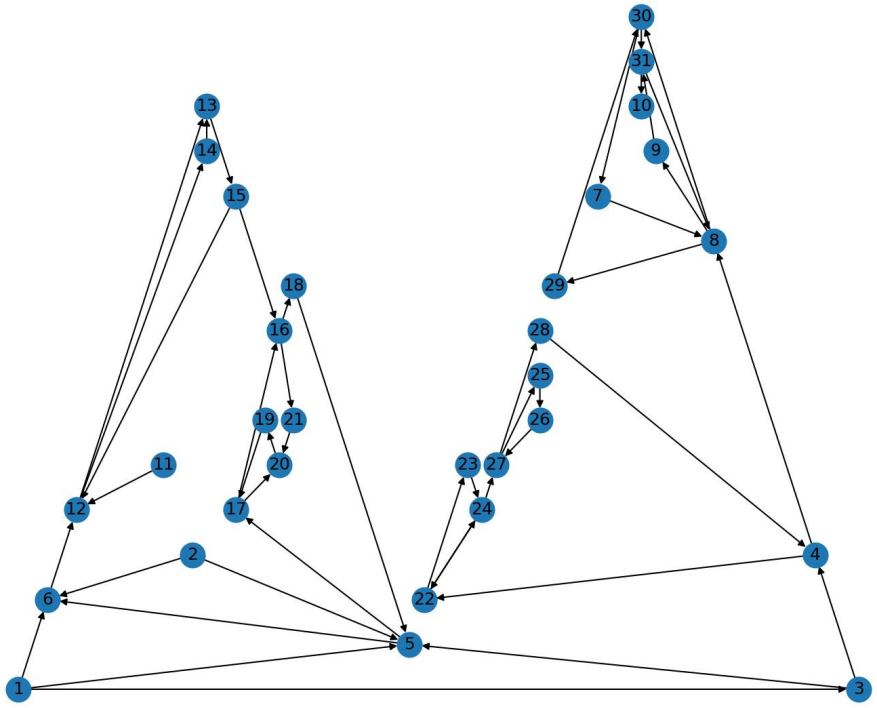

Figure 8: Graf z cyklem skierowanym

2.3 Wyniki

Table 5: BFS

nr grafu	wynik
1	26-> 25-> 23-> 24-> 27-> 28-> 22-> 19-> 11-> 2-> 1-> 6-> 12-> 14-> 13-> 15-> 3-> 5-> 17-> 16-> 21-> 18-> 20-> 4-> 8-> 9-> 29-> 30-> 7-> 31-> 10
2	Graf zawiera skierowany cykl

2.4 Czasy

Table 6: Sortowanie topologiczne

nazwa	skierowany	|V| + |E|	czas [μs]
g2a-1.txt	skierowany	49	2028
g2b-1.txt	skierowany	50	1636
g2a-2.txt	skierowany	361	3402
g2b-2.txt	skierowany	362	2918
g2a-3.txt	skierowany	6241	3703836
g2b-3.txt	skierowany	6242	21884
g2a-4.txt	skierowany	39601	225271
g2b-4.txt	skierowany	39602	165133
g2a-5.txt	skierowany	638401	10502435
g2b-5.txt	skierowany	638402	6528563
g2a-6.txt	skierowany	3996001	67542464
g2b-6.txt	skierowany	3996002	44771434

3 Silnie spójne składowe

3.1 Algorytm

Zastosowałem wersję algorytmu Tarjana do znajdowania silnie spójnych składowych w grafie skierowanym.

- Inicjuję licznik indeksów wierzchołków $index = 0$, a dla każdego wierzchołka ustawiam $index[v] = -1$ oraz $lowlink[v] = -1$.
- Przechodzę przez każdy wierzchołek grafu, a jeżeli $index[v] = -1$, wywołuję procedurę DFS dla wierzchołka v.
- W procedurze DFS ustawiam $index[v]$ oraz $lowlink[v]$ na bieżący indeks, po czym inkrementuję $index$.
- Dodaję v na stos i oznaczam go jako będący na stosie.
- Dla każdego sąsiada w wierzchołka v:
  - Jeżeli w nie był odwiedzony, wywołuję rekurencyjnie DFS dla w, a następnie aktualizuję $lowlink[v] = \min(lowlink[v], lowlink[w])$.
  - Jeżeli w jest na stosie, aktualizuję $lowlink[v] = \min(lowlink[v], index[w])$.
- Po zakończeniu przetwarzania sąsiadów, jeżeli $lowlink[v] = index[v]$, oznacza to początek silnie spójnej składowej:
  - Usuwam wierzchołki ze stosu, aż do v, tworząc nową składową.

Algorytm Tarjana wykonuje tę procedurę dla każdego wierzchołka, odwiedzając każdy tylko raz, dzięki czemu jego złożoność czasowa wynosi $O(V + E)$, gdzie V to liczba wierzchołków, a E to liczba krawędzi w grafie.

3.2 Grafy

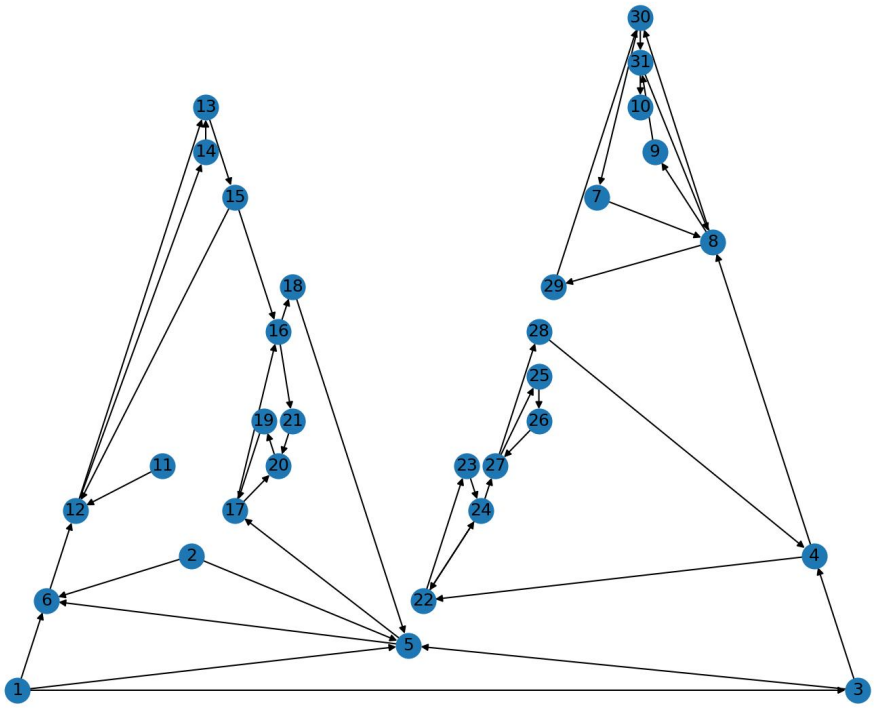

Figure 9: Graf złożony z 8 silnie spójnych składowych

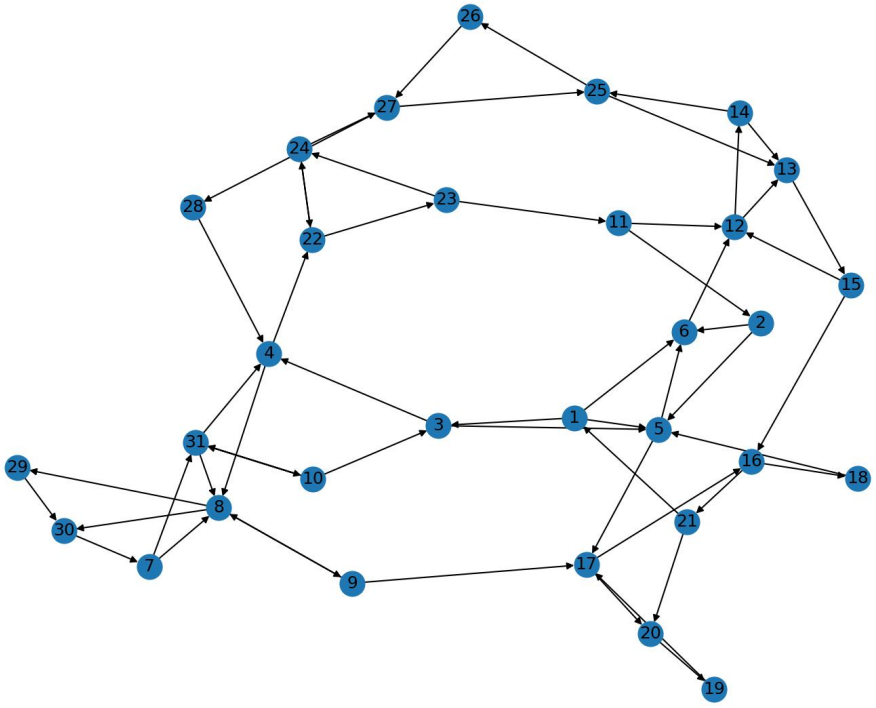

Figure 10: Graf silnie spójny

3.3 Wyniki

Table 7: BFS

nr grafu	wynik
1	Liczba silnie spójnych składowych 8 Rozmiar komponentu 1: 1 10 Rozmiar komponentu 2: 6 9, 7, 31, 30, 29, 8 Rozmiar komponentu 3: 8 28, 26, 25, 27, 24, 23, 22, 4 Rozmiar komponentu 4: 12 14, 17, 19, 20, 21, 18, 16, 15, 13, 12, 6, 5 Rozmiar komponentu 5: 1 3 Rozmiar komponentu 6: 1 1 Rozmiar komponentu 7: 1 2 Rozmiar komponentu 8: 1 11
2	Liczba silnie spójnych składowych 1 Rozmiar komponentu 1: 31 9, 10, 31, 7, 30, 29, 8, 2, 11, 28, 21, 19, 20, 17, 6, 5, 18, 16, 14, 12, 15, 13, 26, 25, 27, 24, 23, 22, 4, 3, 1

3.4 Czasy

Table 8: Silnie spójne składowe

nazwa	skierowany	|V| + |E|	czas [μs]
g3-1.txt	skierowany	55	9250
g3-2.txt	skierowany	292	23920
g3-3.txt	skierowany	2617	4544626
g3-4.txt	skierowany	25952	675758
g3-5.txt	skierowany	259689	10483782
g3-6.txt	skierowany	2598908	109814100

4 Dwudzielność

4.1 Algorytm

Zastosowałem zmodyfikowany algorytm DFS, który koloruje wierchołki na czarno i biało. Dla każdego sukcesora odwiedzonego wierzchołka w algorytmie:

- Jeśli sukcesor v nie był odwiedzony to koloruję go na kolor przeciwny v.
- W przeciwnym wypadku, jeśli kolor sukcesora różni się o koloru v, to graf nie jest dwudzielny.

Złożoność algorytmu jest liniowa, jak DFS, bo zastosowane modyfikacje dodają stałą liczbę operacji.

Dla grafów skierownych, aby algorytm działał poprawnie, należy traktować je jako nieskierowane.

4.2 Grafy

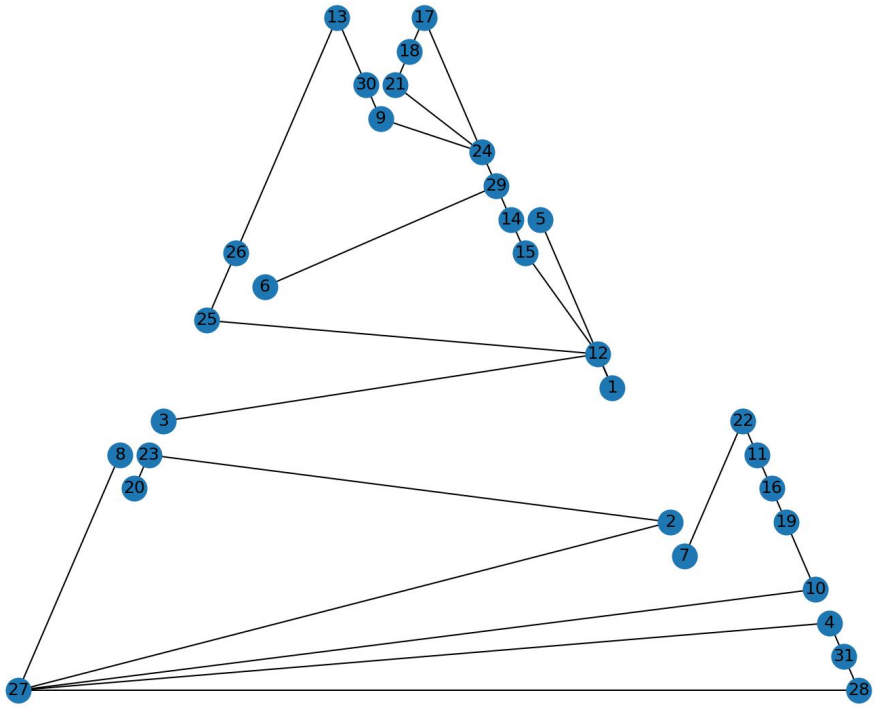

Figure 11: Graf nieskierowany dwudzielny

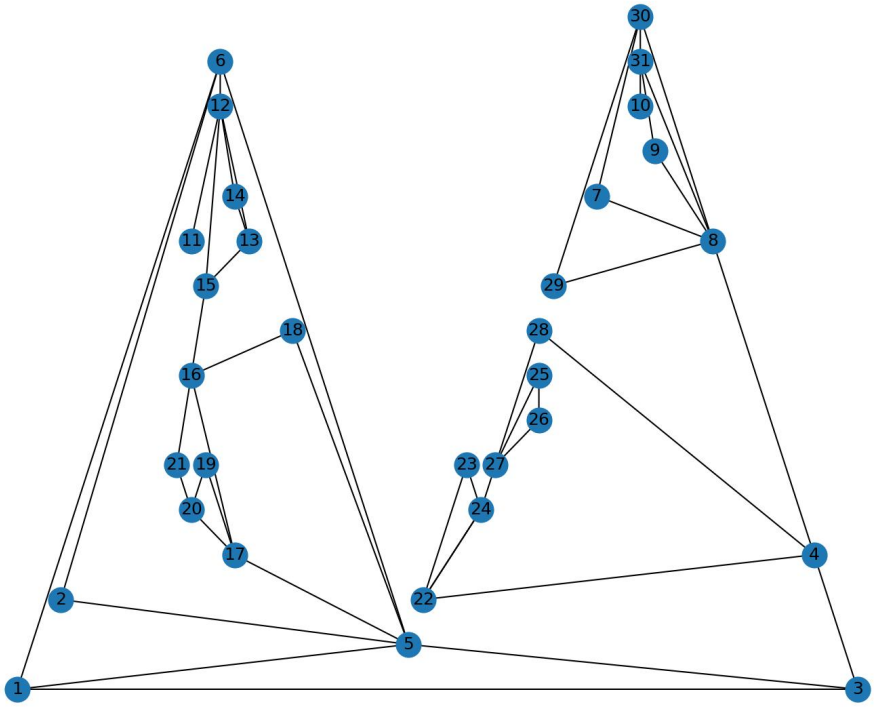

Figure 12: Graf nieskierowany niedwudzielny

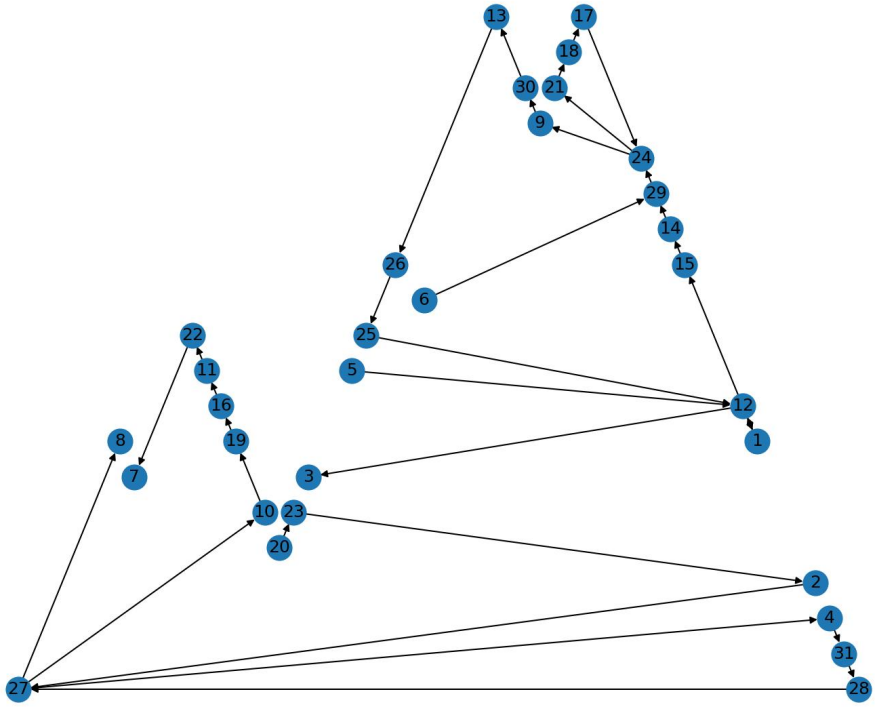

Figure 13: Graf skierowany dwudzielny

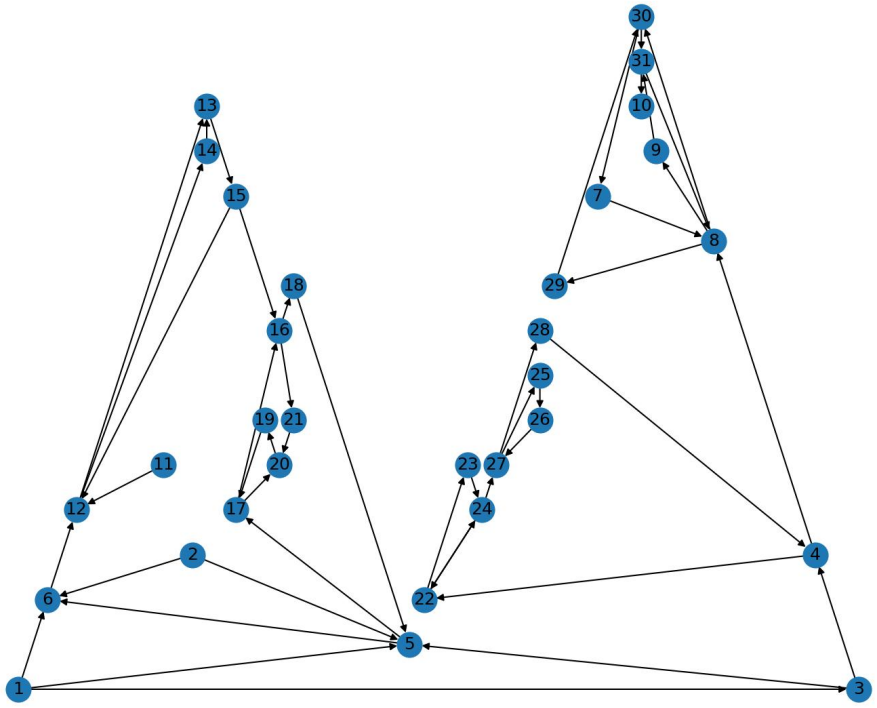

Figure 14: Graf skierowany niedwudzielny

4.3 Wyniki

Table 9: BFS

nr grafu	wynik
1	Wierzchołki niebieskie: 6, 7, 11, 12, 14, 18, 19, 23, 24, 26, 27, 30, 31 Wierzchołki czerwone: 1, 2, 3, 4, 5, 8, 9, 10, 13, 15, 16, 17, 20, 21, 22, 25, 28, 29
2	Graf nie jest dwudzielny
3	Wierzchołki niebieskie: 6, 7, 11, 12, 14, 18, 19, 23, 24, 26, 27, 30, 31 Wierzchołki czerwone: 1, 2, 3, 4, 5, 8, 9, 10, 13, 15, 16, 17, 20, 21, 22, 25, 28, 29
4	Graf nie jest dwudzielny

4.4 Czasy

Table 10: Dwudzielność

nazwa	skierowany	|V| + |E|	czas [μs]
d4a-1.txt	skierowany	40	7816
d4b-1.txt	skierowany	41	5743
u4a-1.txt	nieskierowany	37	3861
u4b-1.txt	nieskierowany	37	2871
d4a-2.txt	skierowany	280	22518
d4b-2.txt	skierowany	281	26752
u4a-2.txt	nieskierowany	317	17342
u4b-2.txt	nieskierowany	317	12263
d4a-3.txt	skierowany	4720	167435
d4b-3.txt	skierowany	4721	137632
u4a-3.txt	nieskierowany	2557	116079
u4b-3.txt	nieskierowany	2557	42739
d4a-4.txt	skierowany	29800	5713498
d4b-4.txt	skierowany	29801	930030
u4a-4.txt	nieskierowany	40957	1163949
u4b-4.txt	nieskierowany	40957	738358
d4a-5.txt	skierowany	479200	26118637
d4b-5.txt	skierowany	479201	20099911
u4a-5.txt	nieskierowany	327677	9742945
u4b-5.txt	nieskierowany	327677	6568739
d4a-6.txt	skierowany	2998000	170479454
d4b-6.txt	skierowany	2998001	111266864
u4a-6.txt	nieskierowany	2621437	79547959
u4b-6.txt	nieskierowany	2621437	53376522