Zadanie 1: Algorytmy DFS i BFS

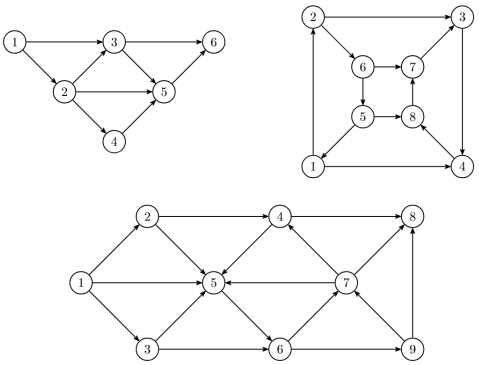
Krzysztof Zając

24 października 2025

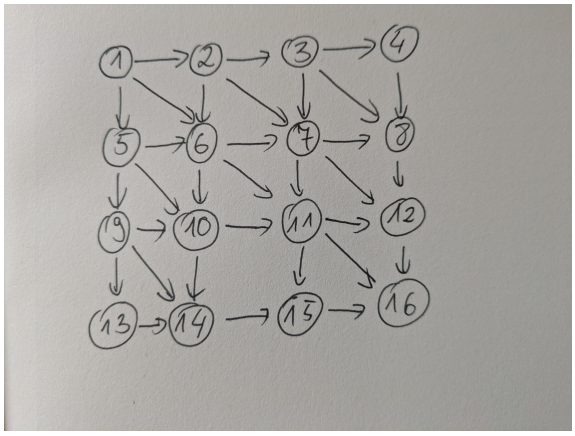
1 Zadanie 1

1.1 Opis problemu

Implementujemy algorytmy DFS i BFS - algorytmy przeszukiwania grafów na poniższych grafach:



Rysunek 1: Grafy do przeszukania w ramach Zadania 1.



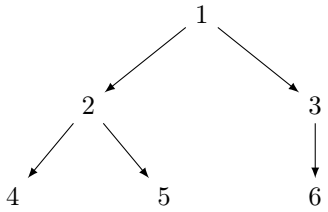
Rysunek 2: Mój graf

1.2 Wyniki

1.3 BFS

1.3.1 06vertices.txt

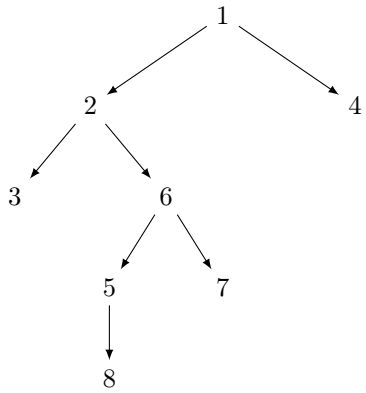
VISIT ORDER: 1 2 3 4 5 6



Rysunek 3: BFS tree for 06vertices.txt

1.3.2 08vertices.txt

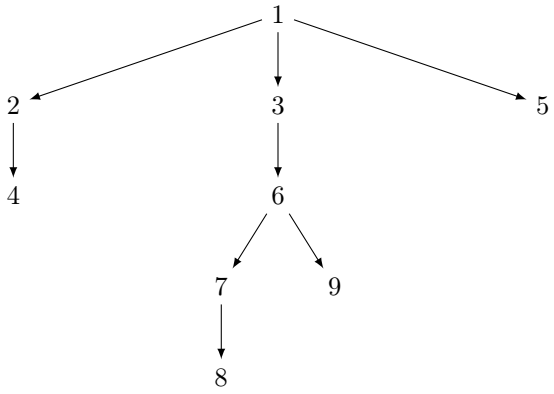
VISIT ORDER: 1 4 2 3 6 5 7 8



Rysunek 4: BFS tree for 08vertices.txt

1.3.3 09vertices.txt

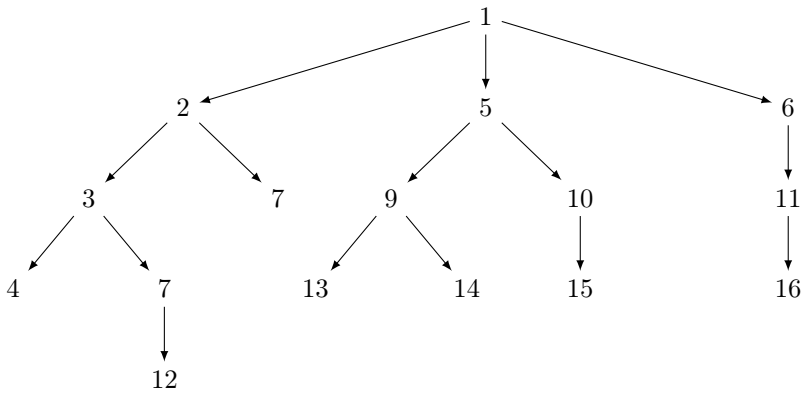
VISIT ORDER: 1 2 3 5 4 6 9 7 8



Rysunek 5: BFS tree for 09vertices.txt

1.3.4 16vertices.txt

VISIT ORDER: 1 2 5 6 3 7 9 10 11 4 8 12 13 14 15 16

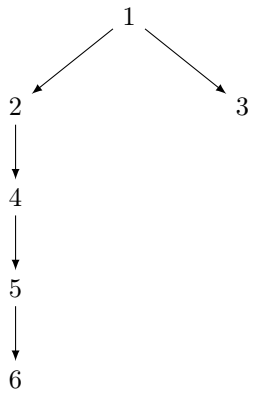


Rysunek 6: BFS tree for 16vertices.txt

1.4 DFS

1.4.1 06vertices.txt

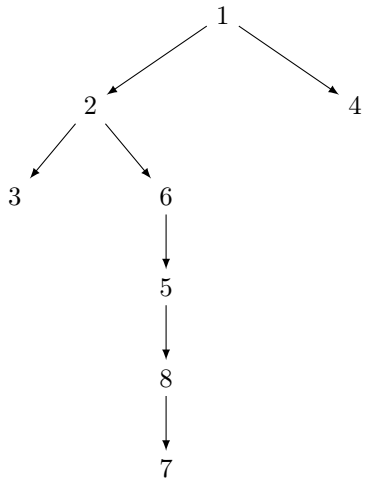
VISIT ORDER: 1 2 4 5 6 3



Rysunek 7: DFS tree for 06vertices.txt

1.4.2 08vertices.txt

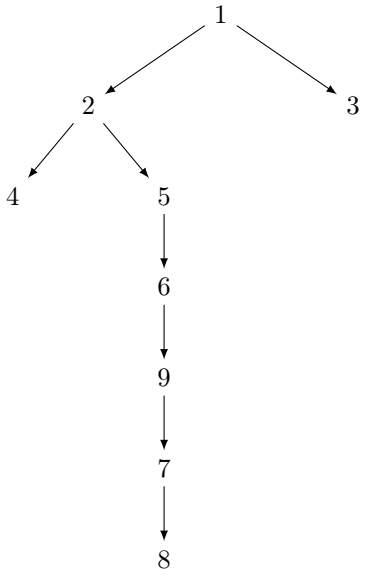
VISIT ORDER: 1 4 2 3 6 5 8 7



Rysunek 8: DFS tree for 08vertices.txt

1.4.3 09vertices.txt

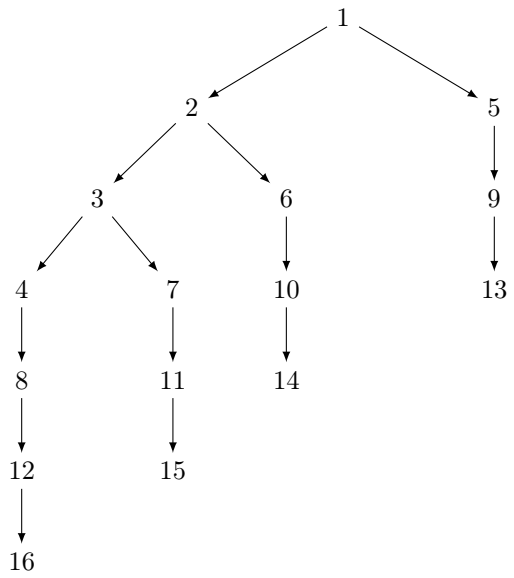
VISIT ORDER: 1 2 4 5 6 9 7 8 3



Rysunek 9: DFS tree for 09vertices.txt

1.4.4 16vertices.txt

VISIT ORDER: 1 2 3 4 8 12 16 7 11 15 6 10 14 5 9 13



Rysunek 10: DFS tree for 16vertices.txt

2 Zadanie 2

2.1 Opis problemu

Implementacja algorytmu sortowania topologicznego

2.2 Wyniki

```
--- Testing 2topological_sort ---

--- Running on g2a-1.txt ---
--- STDOUT ---
ACYCLIC
TOPO_ORDER: 1 2 5 3 6 9 4 7 10 13 8 11 14 12 15 16

--- Running on g2a-2.txt ---
--- STDOUT ---
ACYCLIC
TOPO_ORDER: 1 2 11 3 12 21 4 13 22 31 5 14 23 32 41 6 15 24 33 42 51 7 16 25 34 43 52 61 8 17 26 3

--- Running on g2a-3.txt ---
--- STDOUT ---
ACYCLIC

--- Running on g2a-4.txt ---
--- STDOUT ---
ACYCLIC

--- Running on g2a-5.txt ---
--- STDOUT ---
ACYCLIC

--- Running on g2a-6.txt ---
--- STDOUT ---
ACYCLIC

--- Running on g2b-1.txt ---
--- STDOUT ---
CYCLE

--- Running on g2b-2.txt ---
--- STDOUT ---
CYCLE

--- Running on g2b-3.txt ---
--- STDOUT ---
CYCLE

--- Running on g2b-4.txt ---
--- STDOUT ---
CYCLE

--- Running on g2b-5.txt ---
--- STDOUT ---
CYCLE

--- Running on g2b-6.txt ---
--- STDOUT ---
CYCLE
```

3 Zadanie 3

3.1 Opis problemu

Implementacja algorytmu wykrycia silnie spójnych składowych (tarjan)

3.2 Wyniki

```
--- Testing 3tarjan ---

--- Running on g3-1.txt ---
--- STDOUT ---
SCC_COUNT 5
SCC_SIZES: 1 2 4 4 5
SCC_COMPONENTS:
1: 16
2: 11 10
3: 8 9 7 6
4: 15 14 13 12
5: 5 4 3 2 1

--- Running on g3-2.txt ---
--- STDOUT ---
SCC_COUNT 5
SCC_SIZES: 1 24 36 40 6
SCC_COMPONENTS:
1: 107
2: 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43
3: 28 22 21 27 33 34 35 29 23 17 16 15 14 20 26 32 13 19 25 31 37 38 39 40 41 42 36 30 24 18 12 11
4: 106 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 7
5: 6 5 4 3 2 1

--- Running on g3-3.txt ---
--- STDOUT ---
SCC_COUNT 5
SCC_SIZES: 1 200 400 400 7

--- Running on g3-4.txt ---
--- STDOUT ---
SCC_COUNT 5
SCC_SIZES: 1 2400 3600 4000 8

--- Running on g3-5.txt ---
--- STDOUT ---
SCC_COUNT 5
SCC_SIZES: 1 20000 40000 40000 9

--- Running on g3-6.txt ---
--- STDOUT ---
```

4 Zadanie 4

4.1 Opis problemu

Implementacja algorymtu wykrywającego i badającego dwudzielność

4.2 Wyniki

```
--- Testing 4bipartiteness ---

--- Running on d4a-1.txt ---
--- STDOUT ---
BIPARTITE YES
V0: 1 3 6 8 9 11 14 16
V1: 2 4 5 7 10 12 13 15

--- Running on d4a-2.txt ---
--- STDOUT ---
BIPARTITE YES
V0: 1 3 5 7 9 12 14 16 18 20 21 23 25 27 29 32 34 36 38 40 41 43 45 47 49 52 54 56 58 60 61 63 65
V1: 2 4 6 8 10 11 13 15 17 19 22 24 26 28 30 31 33 35 37 39 42 44 46 48 50 51 53 55 57 59 62 64 66

--- Running on d4a-3.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on d4a-4.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on d4a-5.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on d4a-6.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on d4b-1.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on d4b-2.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on d4b-3.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on d4b-4.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on d4b-5.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on d4b-6.txt ---
--- STDOUT ---
BIPARTITE NO
```

```
--- Running on u4a-1.txt ---
--- STDOUT ---
BIPARTITE YES
V0: 1 4 5 6 7
V1: 2 3 8 9 10 11 12 13 14 15

--- Running on u4a-2.txt ---
--- STDOUT ---
BIPARTITE YES
V0: 1 4 5 6 7 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 64 65 66 67 68 69 70 71 72 73 74 75
V1: 2 3 8 9 10 11 12 13 14 15 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54

--- Running on u4a-3.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on u4a-4.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on u4a-5.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on u4a-6.txt ---
--- STDOUT ---
BIPARTITE YES

--- Running on u4b-1.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on u4b-2.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on u4b-3.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on u4b-4.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on u4b-5.txt ---
--- STDOUT ---
BIPARTITE NO

--- Running on u4b-6.txt ---
--- STDOUT ---
BIPARTITE NO
```