Filip Obuchowicz (S223421), Kamil Pizon (S223434)

**Temat 27: Opracować i zaimplementować algorytm znajdujący najkrótszy cykl w grafie skierowanym.**

**Opis działania aplikacji:**

1. Użytkownik wprowadza strukturę grafu w formie tekstowej (jako słownik Pythona), np. {'A': ['B,E'], 'C': ['D']}, gdzie ‘A’ to nazwa wierzchołka, a ‘B’ i ‘E’ to wierzchołki do których można dojść z wierzchołka ‘A’.
2. Po kliknięciu przycisku, program analizuje wprowadzony graf, wykorzystując algorytm bazujący na wielokrotnym przeszukiwaniu wszerz (BFS).
3. Wyświetlenie wyniku:
   * Tekstowo: Aplikacja informuje o długości znalezionego najkrótszego cyklu oraz wyświetla węzły tworzące ten cykl. Jeśli cykl nie istnieje, również o tym informuje.
   * Wizualnie: Graf jest rysowany na ekranie. Jeśli znaleziono najkrótszy cykl, jego węzły i krawędzie są podświetlane na czerwono dla łatwiejszej identyfikacji.
4. Aplikacja oferuje również opcje załadowania przykładowych grafów oraz wyczyszczenia bieżących danych, co ułatwia testowanie i eksperymentowanie.

**Algorytm Przeszukiwania Wszerz (BFS - Breadth-First Search)**

1. **Start:** Zaczyna od węzła początkowego.
2. **Sąsiedzi:** Odwiedza wszystkich jego **bezpośrednich sąsiadów** (pierwsza "warstwa").
3. **Kolejni sąsiedzi:** Następnie odwiedza sąsiadów tych sąsiadów (druga "warstwa"), którzy nie byli jeszcze odwiedzeni.
4. **Kontynuacja:** Proces ten jest kontynuowany warstwa po warstwie, aż wszystkie osiągalne węzły zostaną odwiedzone.

**Kluczowe cechy:**

* Używa **kolejki** do śledzenia, które węzły odwiedzić następnie.
* Znajduje **najkrótszą ścieżkę** (w sensie liczby krawędzi) od węzła startowego do wszystkich innych osiągalnych węzłów (w grafach nieważonych).
* Jest kompletny (jeśli ścieżka istnieje, znajdzie ją) i optymalny (znajduje najkrótszą).

Złożoność algorytmu

* **Złożoność czasowa: O(V^3 + V\*E)** (lub upraszczając, często podaje się O(V^3) dla grafów gęstych i rzadkich w tym scenariuszu).
* **Złożoność pamięciowa: O(V^2 + E)** (lub upraszczając O(V^2)).