# Dokumentacja Programu

## 1. Wstęp

Program służy do odnalezieniea i wizualizacji cykli Hamiltona na grafie reprezentowanym przez okręgi umieszczone na dużym okręgu na podstawie maceirzy sąsiedztwa.

## 2. Wymagania

Wymagane biblioteki: tkinter, math.

## 3. Struktura Programu

Program składa się z komponentów do tworzenia interfejsu użytkownika oraz logiki do rysowania okręgów i znajdowania cykli Hamiltona.

## 4. Opis Funkcji

## `create\_circles\_on\_circle(x, y, z)`

- Parametry:

- `x` (int): Promień dużego okręgu.

- `y` (int): Liczba mniejszych kół.

- `z` (list): Macierz sąsiedztwa.

- Zadanie: Rysuje mniejsze koła i łączy je zgodnie z macierzą sąsiedztwa.

## `isSafe(v, graph, path, pos)`

- Parametry:

- `graph` (lista list): Macierz sąsiedztwa

- `pos`: aktualna pozycja w ścieżce

- `path` : Obecna ścieżka

- `v`: wierzchołek w ścieżce

- Zadanie: Sprawdza, czy można dodać wierzchołek v do ścieżki. Warunki bezpieczeństwa to: musi istnieć krawędź z ostatniego wierzchołka w ścieżce do v, a v nie może być już w ścieżce (aby uniknąć powtórzeń).

## `hamCycle(graph)`

- Parametry:

- `graph` (lista list): Macierz sąsiedztwa

- Zadanie: Funkcja inicjuje proces szukania cyklu Hamiltona. Tworzy ścieżkę (path), zaczynając od wierzchołka 0, i ustawia wszystkie wierzchołki jako nieodwiedzone (visited).Następnie wywołuje funkcję FindHamCycle, przekazując jej graf, ścieżkę, oraz informacje o odwiedzonych wierzchołkach

## `FindHamCycle(graph, pos, path, visited)`

- Parametry:

- `graph` (lista list): Macierz sąsiedztwa

- `pos`: aktualna pozycja w ścieżce

- `path` : Obecna ścieżka

- `visited` (lista): Lista zawierająca informacje czy dany wierzchołek był już odwiedzony

- Zadanie: Jest to rekurencyjna funkcja, która próbuje budować cykl Hamiltona krok po kroku. Sprawdza, czy wszystkie wierzchołki zostały odwiedzone. Jeśli tak, sprawdza, czy istnieje połączenie między ostatnim wierzchołkiem na ścieżce a pierwszym wierzchołkiem, zamykając tym samym cykl. Jeśli cykl jest kompletny i poprawny, dodaje go do globalnej listy cycles. Jeśli cykl nie jest jeszcze kompletny, funkcja próbuje dodać kolejny wierzchołek do ścieżki, sprawdzając za pomocą funkcji isSafe, czy jest to możliwe.

## `draw\_Hamilton(x)`

- Parametry:

- `x` (int): Cykl Hamiltona.

- Zadanie: Rysuje jeden z cykli Hamiltona.

## `delet\_previous()`

- Zadanie: Usuwa zaznaczony cykl Hamiltona.

## `on\_button\_click()`

- Zadanie: Przełącza wyświetlane cykle Hamiltona.

## `has\_cycle()`

-Zadanie: Wypisuje informacje o cyklach Hamiltona.

## 5. Interfejs Użytkownika

W lewym górnym rogu wyświetlana jest ilość cykli Hamiltona, w podanym Grafie. Przycisk Next pozwala poruszać się miedzy nimi.

## 6. Przykładowe Użycie

Po uruchomieniu programu wyświetlany jest stworzony na podstawie macierzy(zadanej w kodzi) graf oraz wyświetlana liczba cykli Hamiltona. Jeśli jest ona większa od 0 to po wciśnięciu przycisku Next, zostanie wyświetlony pierwszy cykl. W przypadku większej ilości cykli przycisk Next wyświetla je w koło, wyjście następuje poprzez zamknięcie okna.

## 7. Załączniki

Obraz zawierający linia, krąg, diagram, zrzut ekranu

Opis wygenerowany automatycznieObraz zawierający tekst, linia, diagram, zrzut ekranu

Opis wygenerowany automatycznie