

## Instrukcja obsługi:

- Generowanie danych:
  - `g++ create_data.cpp -o create`
  - `./create [ilość_wierzchołków] [szansa] [typ_pliku] [nazwa_pliku]`
    - Ilość wierzchołków – ilość wierzchołków generowanego grafu
    - Szansa – szansa na wystąpienie krawędzi
    - Typ pliku – typ pliku, do którego zapisujemy macierz sąsiedztwa. Możliwe opcje to „b”, „t” lub „bt”, gdzie t i b oznaczają odpowiednio plik tekstowy (.txt) i binarny (.bin)
    - Nazwa pliku – nazwa pliku bez rozszerzenia, do którego zostanie zapisana macierz.
- Algorytm Dijkstry:
  - `mpicxx Dijkstra.cpp -o dijkstra`
  - `mpiexec -f nodes -n <ilość procesów> ./dijkstra [typ pliku] [plik] [źródło] [cel]`
    - Typ pliku – rozszerzenie pliku, z którego będzie czytana macierz sąsiedztwa. Może być „b” do plików binarnych (.bin) lub „t” do plików tekstowych (.txt)
    - Plik – nazwa pliku do wczytania
    - Źródło – numer wierzchołka, z którego będzie liczony algorytm Dijkstry.
    - Ujście – numer wierzchołka, do którego algorytm Dijkstry policzy najkrótszą ścieżkę i ją wypisze. Przy podaniu wartości „-1” algorytm policzy najkrótsze ścieżki do wszystkich wierzchołków i zapisze je do pliku „path.txt”.
- Rysowanie grafu i ścieżek:
  - Graf jest zapisywany do obrazu „graph.png”
  - `python3 draw_graph.py [plik_tekstowy]`
    - Plik tekstowy – nazwa pliku tekstowego z macierzą sąsiedztwa grafu.
    - Opcje:
      - `-s [--shortest-path]` – po podaniu numeru wierzchołka zawierającego się w grafie i nie będącego źródłem, zaznacza wybraną ścieżkę na obrazie. Domyślnie równa -1.
      - `-i [--image-size]` – wielkość figury, w której rysowany jest graf. Domyślnie równa 10.