

## Format danych wejściowych

Obsługiwany format danych wejściowych to pliki tekstowe opisujące jeden lub więcej grafów. W pierwszej linii pliku podana jest liczba grafów do przetworzenia. W kolejnych liniach są opisy grafów, każdy z opisów pojedynczego grafu oddzielony jest jedną wolną linią. Opis pojedynczego grafu to:

- Liczba wierzchołków (ozn.  $n$ ) oddzielona nową linią
- W kolejnych  $n$  liniach podane jest  $n$  wartości macierzy sąsiedztwa (wartości to 0 lub 1, oznaczające brak lub obecność krawędzi skierowanej). Wartości w jednej linii oddzielone są między sobą spacją

Dane wejściowe znajdują się w folderze Examples:

- 6\_wierzchołkow\_graf1 - mały graf o 6 wierzchołkach,
- 6\_wierzchołkow\_graf3 - mały graf o 6 wierzchołkach,
- 7\_wierzchołkow\_graf2 - mały graf o 7 wierzchołkach,
- 16\_wierzchołkow\_graf4 - duży graf o 16 wierzchołkach (przeznaczony dla algorytmów aproksymacyjnych i zalecana opcja output: *short*),
- 6\_wierzchołkow\_graf1 - duży graf o 30 wierzchołkach (przeznaczony dla algorytmów aproksymacyjnych i zalecana opcja output: *short*),
- wiele\_grafow - 9 grafów (przeznaczone dla algorytmów aproksymacyjnych, zalecana opcja output: *short*).

Jeżeli wykonanie programu będzie trwać zbyt długo, to można zrezygnować z wykonywania algorytmu wprowadzając kombinację klawiszy "Ctrl + C".

## Instrukcja uruchomienia programu z pliku wykonywalnego exe

Do uruchomienia programu dostarczyliśmy skompilowany przez nas program exe (znajduje się w folderze Exe w rozpakowanym archiwum zip). Aby wygodnie podawać nazwy plików zawierających dane wejściowe przy uruchamianiu programu należy umieścić pliki z folderu Examples w folderze, w którym znajduje się program (taio\_graph.exe), czyli w folderze Exe. Program został skompilowany na 64-bitowym procesorze w systemie Windows. Wszystkie algorytmy dostępne są w jednym programie taio\_graph.exe. Program uruchamia się z konsoli (np. command prompt na Windows). Poniżej są instrukcje uruchomienia każdego z algorytmów z jego opcjami. Można uruchomić tylko jeden algorytm na raz, tzn. dla wykonania wielu algorytmów konieczne jest ponowne wywołanie programu z innymi parametrami. Poniżej są instrukcje dla algorytmów dostępnych w ramach naszego rozwiązania.

## Wyznaczanie rozmiaru grafu

Format dla uruchomienia algorytmu liczenia rozmiaru grafu (jest tylko opcja dokładna) to:

```
taio_graph.exe (-s|--size) exact -f nazwa_pliku.txt [-o|--output long|short]
```

- -s exact - opcja krótka uruchomienia programu liczenia rozmiaru grafu
- --size exact - opcja długa uruchomienia programu liczenia rozmiaru grafu (przed size są dwie półpauzy "-")
- -f nazwa\_pliku.txt - podanie pliku wejściowego dla algorytmu liczenia rozmiaru. Plik może zawierać jeden lub więcej grafów, rozmiar zostanie obliczony dla każdego z nich.
- -o, --output long|short - opcjonalny argument określający sposób wyświetlania wyników: long (długi - macierz sąsiedztwa i rozmiar grafu) lub short (krótki - sam rozmiar). Domyślna wartość to long.

**Przykład użycia** Poniżej dwa przykłady użycia (wywołanie z opcją krótką "-s" i długą "--size")

```
taio_graph.exe -s exact -f data.txt --output short
```

```
taio_graph.exe --size exact -f data.txt
```

## Wyznaczenie odległości dwóch grafów

Format dla uruchomienia algorytmu wyznaczenia odległości między dwoma zadanymi grafami według opracowanej przez nas metryki to:

```
taio_graph.exe (-d|--distance) exact|approx|both  
                -f1 nazwa_pliku1.txt  
                -f2 nazwa_pliku2.txt  
                [-o|--output long|short]
```

- -d exact|approx|both - opcja krótka uruchomienia programu liczenia odległości
- --distance exact|approx|both - opcja długa uruchomienia programu liczenia odległości (przed size są dwie półpauzy "-")
- exact|approx|both - po opcji wyboru algorytmu -d lub --distance należy podać po spacji wybór wersji algorytmu odpowiednio **exact** to algorytm dokładny, **approx** to algorytm aproksymacyjny, a **both** to uruchomienie obu wersji.
- -f1 nazwa\_pliku1.txt - podanie pierwszego pliku wejściowego dla algorytmu liczenia odległości dwóch grafów. Do obliczeń zostanie użyty **tylko pierwszy** graf z pliku.

- -f2 nazwa\_pliku2.txt - podanie drugiego pliku wejściowego dla algorytmu liczenia odległości dwóch grafów. Do obliczeń zostanie użyty **tylko pierwszy** graf z pliku.
- -o, --output long|short - opcjonalny argument określający sposób wyświetlania wyników: long (długi - macierze sąsiedztwa i odległość) lub short (krótki - sama odległość). Domyślna wartość to long.

**Przykład użycia** Poniżej przykłady użycia dla algorytmu liczenia odległości

```
taio_graph.exe -d both -f1 data1.txt -f1 data2.txt
```

```
taio_graph.exe --distance approx -f1 data1.txt -f1 data2.txt -o short
```

```
taio_graph.exe --distance exact -f1 data1.txt -f1 data2.txt --output long
```

## Wyznaczanie najdłuższego cyklu i liczby najdłuższych cykli

Format dla uruchomienia algorytmu wyznaczania najdłuższego cyklu i liczby najdłuższych cykli to:

```
taio_graph.exe (-c|--cycle) exact|approx|both -f nazwa_pliku.txt  
[-o|--output long|short]
```

- -c exact|approx|both - opcja krótka uruchomienia programu wyznaczania najdłuższego cyklu i liczby najdłuższych cykli
- --cycle exact|approx|both - opcja długa uruchomienia programu wyznaczania najdłuższego cyklu i liczby najdłuższych cykli (przed cycle są dwie półpauzy "-")
- exact|approx|both - po opcji wyboru algorytmu -d lub --distance należy podać po spacji wybór wersji algorytmu odpowiednio **exact** to algorytm dokładny, **approx** to algorytm aproksymacyjny, a **both** to uruchomienie obu wersji.
- -f nazwa\_pliku.txt - podanie pliku wejściowego dla algorytmu liczenia najdłuższego cyklu i liczby najdłuższych cykli. Plik może zawierać jeden lub więcej grafów, wyniki zostaną podane dla każdego z nich.
- -o, --output long|short - opcjonalny argument określający sposób wyświetlania wyników: long (długi - macierze sąsiedztwa, rozmiar cyklu, wypisywane wierzchołki najdłuższego cyklu oraz zaznaczony przykładowy cykl w macierzy sąsiedztwa) lub short (krótki - rozmiar cyklu, wypisywane wierzchołki najdłuższego cyklu). Domyślna wartość to long.

**Przykład użycia** Poniżej przykłady użycia dla programu wyznaczania najdłuższego cyklu i liczby najdłuższych cykli:

```
taio_graph.exe -c both -f data.txt
```

```
taio_graph.exe --cycle approx -f data.txt
```

```
taio_graph.exe --cycle exact -f data.txt
```

## Wyznaczanie minimalnego rozszerzenia hamiltonowskiego i liczby cykli hamiltona

Format dla uruchomienia algorytmu wyznaczania minimalnego rozszerzenia hamiltonowskiego i liczby cykli hamiltona:

```
taio_graph.exe (-h|--hamilton) exact|approx|both -f nazwa_pliku.txt  
                [-r wartosc_retry_factor]  
                [-o|--output long|short]
```

- -h exact|approx|both - opcja krótka uruchomienia programu wyznaczania minimalnego rozszerzenia grafu do grafu zawierającego cykl Hamiltona i liczby cykli Hamiltona
- --hamilton exact|approx|both - opcja długa uruchomienia wyznaczania minimalnego rozszerzenia hamiltonowskiego i liczby cykli hamiltona (przed hamilton są dwie półpauzy "-")
- exact|approx|both - po opcji wyboru algorytmu -d lub -distance należy podać po spacji wybór wersji algorytmu odpowiednio **exact** to algorytm dokładny, **approx** to algorytm aproksymacyjny, a **both** to uruchomienie obu wersji.
- -f [nazwa\_pliku] - podanie pliku wejściowego dla algorytmu wyznaczającego minimalne rozszerzenie hamiltonowskie i liczbę cykli Hamiltona. Plik może zawierać jeden lub więcej grafów, wyniki zostaną podane dla każdego z nich.
- -r wartość retry\_factor - opcjonalny argument uruchomienia ustawiający parametr wykonania algorytmu aproksymacyjnego nazwany retry factor. Podstawowo wartość ustawiona jest na 10, a akceptowalne wartości są od 1 do 15 włącznie.
- -o, --output long|short - opcjonalny argument określający sposób wyświetlania wyników: long (długi - dla każdego grafu: macierz sąsiedztwa, dla wersji aproksymacyjnej wartość retry factor, rozmiar najmniejszego rozszerzenia, krawędzie najmniejszego rozszerzenia, liczba cykli Hamiltona w rozszerzonym grafie (czyli grafie do którego dodano krawędzie z najmniejszego znalezionej rozszerzenia), wszystkie cykle Hamiltona w rozszerzonym grafie oraz macierze sąsiedztwa, na których zaznaczono cykle) lub short (krótki - dla każdego grafu: dla wersji aproksymacyjnej wartość parametru retry factor, rozmiar najmniejszego rozszerzenia, krawędzie najmniejszego rozszerzenia, liczba cykli Hamiltona w rozszerzonym grafie, przykładowy cykl Hamiltona w rozszerzonym grafie). Domyślna wartość to long.

**Przykład użycia** Poniżej przykłady użycia dla programu wyznaczania minimalnego rozszerzenia hamiltonowskiego i liczby cykli hamiltona:

```
taio_graph.exe -h both -f data.txt -o short
```

```
taio_graph.exe --hamilton approx -f data.txt -r 13 --output long
```

```
taio_graph.exe --hamilton exact -f data.txt
```

```
taio_graph.exe -h both -f data.txt -r 7 --output short
```

# Instrukcja kompilacji projektu z plików źródłowych

Dodatkowo oprócz wersji wykonywalnej programu możliwe jest samodzielne skompilowanie projektu. Do tego na docelowym komputerze potrzebny jest visual studio oraz system cmake. Po zbudowaniu projektu plik wykonywalny powinien znajdować się w odpowiednim subfolderze projektu (opisane w poniższych instrukcjach). Korzystanie z tak skompilowanego projektu jest identyczne jak z prekompilowanej wersji (do której instrukcja jest powyżej).

## Kompilacja z Visual Studio

Do skompilowania programu z Visual Studio należy:

1. otworzyć folder projektu przez Visual Studio,
2. w panelu przeglądu solucji (ang. Solution Explorer, zwykle po prawej stronie widoku aplikacji) otworzyć plik CMakeLists.txt,
3. wcisnąć kombinację Ctrl + S,
4. w górnym menu jako opcję uruchomienia wybrać taio\_graph.exe i następnie wcisnąć Ctrl + B (kompilacja projektu).

Plik wykonywalny można znaleźć w **out/build/x64-Debug/taio\_graph.exe** (ścieżka relatywna względem folderu głównego projektu czyli tego w którym bezpośrednio jest plik CMakeLists.txt)

## Kompilacja z CMake z konsoli

Do skompilowania za pomocą narzędzia CMake potrzebna jest konsola. Kolejno należy:

1. Otworzyć folder projektu w konsoli
2. utworzyć folder o nazwie build

```
mkdir build
```

3. przejść do folderu build

```
cd build
```

4. stworzyć konfigurację do kompilacji za pomocą komendy

```
cmake -S .. -B .
```

5. skompilować aplikację za pomocą komendy:

```
cmake --build .
```

Plik wykonywalny w tym przypadku jest w **build/Debug/taio\_graph.exe** (gdzie build to utworzony przez nas folder w opisanym tutaj procesie kompilacji).