

### **Część programistyczna**

Należy zaimplementować algorytmy rozwiązujące problem cyklu Eulera oraz problem cyklu Hamiltona w grafie nieskierowanym oraz digrafie. Dla obu problemów należy zaimplementować:

1. Algorytm rozwiązujący wersję decyzyjną. Algorytm DEC dla cyklu Eulera powinien sprawdzać czy graf spełnia warunek konieczny i wystarczający. Algorytm DHC dla cyklu Hamiltona w grafie nieskierowanym powinien sprawdzać warunek wystarczający z twierdzenia Orego, natomiast w grafie skierowanym – warunek wystarczający z twierdzenia Woodalla.
2. Algorytm rozwiązujący wersję przeszukiwania wykorzystujący technikę powracania. Algorytm SEC dla cyklu Eulera powinien realizować przeszukiwanie przestrzeni rozwiązań, polegające na wyborze kolejnych krawędzi i cofaniu się w przypadku braku możliwości kontynuacji cyklu. Algorytm SHC dla cyklu Hamiltona powinien stosować metodę Robertsa-Floresa.

Algorytmy operujące na grafach nieskierowanych należy zaimplementować w taki sposób, aby przetwarzały grafy zapisane w macierzy sąsiedztwa, natomiast algorytmy operujące na grafach skierowanych mają działać na macierzy grafu. Nie należy przechowywać tej samej informacji o grafie w więcej niż jednej reprezentacji maszynowej grafu.

Program powinien:

1. Być odpornym na błędy wprowadzane przez użytkownika.
2. Generować losowy graf nieskierowany, losowy graf skierowany lub wczytywać graf z pliku tekstowego (pierwsza linia: liczba wierzchołków  $n$  i liczba krawędzi/łuków  $m$ , kolejne linie: pary wierzchołków  $(u, v)$ , wierzchołki numerowane od 1).
3. W trybie demonstracyjnym umożliwiać wybór problemu (cykl Eulera lub Hamiltona), wersji problemu (decyzyjna lub przeszukiwania), oraz wybór typu grafu (nieskierowany lub skierowany).
4. Dla każdego uruchomienia algorytmu podawać czas wykonania.
5. Dla każdego uruchomienia algorytmu rozwiązującego wersję przeszukiwania podawać liczbę operacji cofania (backtracków).
6. W trybie demonstracyjnym wyświetlać: graf w odpowiedniej reprezentacji maszynowej, znalezione rozwiązanie, w wersji przeszukiwania także długość rozwiązania.

### **Część eksperymentalna**

Generowanie danych: dla każdej liczby wierzchołków  $n$  należy wygenerować grafy nieskierowane oraz grafy skierowane o różnym poziomie nasycenia  $s$ , rozumianym jako stosunek liczby krawędzi do maksymalnej liczby krawędzi/łuków w odpowiednim grafie  $n$  wierzchołkach.

Należy wybrać 10–15 różnych wartości  $n$  rosnących liniowo (np.: 1000, 2000, 3000, 4000, ... dla problemu poszukiwania cyklu Eulera oraz odpowiednio mniejsze dla problemu poszukiwania cyklu Hamiltona). Zakres należy dobrać tak, aby możliwe było zaobserwowanie wzrostu czasu obliczeń wraz ze wzrostem  $n$  oraz aby pomiary były możliwe do wykonania w rozsądnym czasie. Dla każdego  $n$ , należy przeprowadzić eksperymenty obliczeniowe dla grafów o różnym nasyceniu  $s$ .

Procedura pomiarowa:

1. dla każdego z czterech algorytmów (DEC, DHC, SEC, SHC), liczby wierzchołków  $n$  oraz nasycenia  $s = [10, 20, 30, 40, 50, 60, 70, 80, 90]$ , należy wygenerować 10 niezależnych grafów i zmierzyć czas poszukiwania rozwiązania bez wcześniejszego sprawdzania (algorytmem DEC/DHC), czy graf zawiera dany cykl – osobno dla grafów nieskierowanych i digrafów.
2. dla każdego z dwóch algorytmów rozwiązujących problem przeszukiwania (SEC, SHC), liczby wierzchołków  $n$  oraz nasycenia  $s = [50, 60, 70, 80, 90]$ , należy wygenerować 10 niezależnych grafów eulerowskich (dla SEC) i 10 niezależnych grafów hamiltonowskich (dla SHC) oraz zmierzyć czas poszukiwania rozwiązania w tych grafach – osobno dla grafów nieskierowanych i digrafów.

Każdy punkt pomiarowy ma być średnią z 10 uruchomień algorytmu. Dla każdego punktu pomiarowego należy obliczyć odchylenie standardowe. Wszystkie wyniki należy zapisać w tabelach.

### Sprawozdanie

Sprawozdanie powinno zawierać:

1. Wykresy liniowe przedstawiające zależność średniego czasu obliczeń  $t$  (oś  $Y$ ) od liczby wierzchołków  $n$  (oś  $X$ ) dla grafów o średnim nasyceniu ( $s = 50\%$ ) zawierających cykle. Należy przygotować 4 wykresy funkcji  $t=f(n)$ : dwa dla wersji decyzyjnej problemu (oddzielnie dla DEC i DHC), dwa dla wersji przeszukiwania (oddzielnie dla SEC i SHC). Na każdym wyrysować 2 krzywe – jedną dla grafu nieskierowanego, a drugą dla grafu skierowanego. Odchylenia standardowe należy zaznaczyć jako słupki błędów.
2. Wykresy liniowe przedstawiające zależność średniego czasu obliczeń  $t$  (oś  $Y$ ) od liczby wierzchołków  $n$  (oś  $X$ ) dla grafów o średnim nasyceniu ( $s = 50\%$ ), które mogą zawierać lub nie zawierać cykli. Należy przygotować 4 wykresy funkcji  $t=f(n)$ : dwa dla wersji decyzyjnej problemu (oddzielnie dla DEC i DHC), dwa dla wersji przeszukiwania (oddzielnie dla SEC i SHC). Na każdym wyrysować 2 krzywe – jedną dla grafu nieskierowanego, a drugą dla grafu skierowanego. Odchylenia standardowe należy zaznaczyć jako słupki błędów.
3. Wykresy powierzchniowe 3D  $t=f(n,s)$  zależności czasu obliczeń  $t$  od liczby  $n$  wierzchołków w grafie i nasycenia  $s$ . Należy przygotować 4 wykresy funkcji  $t=f(s)$ : jeden dla algorytmu SEC i grafu nieskierowanego, jeden dla algorytmu SEC i grafu skierowanego, jeden dla algorytmu SHC i grafu nieskierowanego, jeden dla algorytmu SHC i grafu skierowanego.
4. Analizę wyników eksperymentów obliczeniowych. Należy przeanalizować wyniki eksperymentów dla wszystkich algorytmów i reprezentacji grafu i odpowiedzieć na pytania:
  - Jak różni się liczba backtracków w obu problemach? Dlaczego zastosowanie tej samej techniki (backtracking) prowadzi do tak różnych wyników dla problemu Eulera i Hamiltona?
  - Jak zmienia się czas działania wraz ze wzrostem  $n$ ?
  - Jak zmiana nasycenia grafu wpływa na liczbę operacji cofania w algorytmie SHC? Czy łatwiej znaleźć cykl Hamiltona w grafie rzadkim czy gęstym?
  - Dla jakich typów grafów (lub parametrów  $n$ ,  $s$ ) algorytmy z powracaniem wykazywały największe odchylenie standardowe czasu obliczeń?
  - Czy problem Eulera rzeczywiście jest łatwy w praktyce?
  - W którym momencie algorytm Hamiltona przestaje być wykonywalny?
  - Czy zauważasz różnicę w czasie poszukiwania rozwiązania w grafach zawierających cykle i grafach, które mogą ich nie zawierać?
  - Czy narzucona reprezentacja maszynowa grafu ułatwiła czy utrudniła implementację konkretnych algorytmów (np. sprawdzanie spójności lub wyszukiwanie sąsiadów)?
5. Analizę złożoności obliczeniowej. Wyjaśnij różnicę między klasą złożoności problemu a złożonością rozwiązującego go algorytmu. Dlaczego problem cyklu Eulera należy do klasy P? Dlaczego problem cyklu Hamiltona jest silnie NP-trudny/NP-zupełny? Czy istnieją grafy hamiltonowskie, które nie spełniają twierdzenia Orego/Woodalla?
6. Wnioski końcowe. W podsumowaniu należy porównać oba problemy, opisać wpływ zastosowanego podejścia (backtracking) do rozwiązania wersji przeszukiwania, przedyskutować praktyczną wykonalność algorytmów, wskazać najbardziej efektywną reprezentację dla każdego algorytmu, określić kompromisy pamięć vs czas.