

# Projektowanie Algorytmów i Metody Sztucznej Inteligencji

## Projekt 3

### **UWAGA1:**

Nie należy stosować gotowych kontenerów STL do implementacji zadań (nie dotyczy gry).

### **UWAGA2:**

Wszystkie struktury danych oraz algorytmy należy implementować zgodnie z opisem na wykładzie.

### **UWAGA3:**

Zadania poświęcone badaniu efektywności algorytmów grafowych zakładają badanie efektywności w zależności od metody reprezentacji grafów. Należy uwzględnić reprezentacje grafu w postaci macierzy sąsiedztwa oraz listy sąsiedztwa. Badania należy wykonać dla 5 różnych liczb wierzchołków w grafie  $V$  (np. 10, 50, 100, 500 i 1000) oraz następujących gęstości grafu: 25%, 50%, 75% oraz dla grafu pełnego. Dla każdego zestawu parametrów: algorytm, reprezentacja grafu, liczba wierzchołków i gęstość grafu należy wygenerować po 100 losowych instancji, natomiast w sprawozdaniu umieścić wyniki uśrednione.

## Zadania

### Zadania na ocenę dst (3.0):

1. Należy zaimplementować tablicę haszującą przechowującą  $N$  integerów z wykorzystaniem funkcji haszującej w postaci  $x\%N$ , gdzie  $x$  reprezentuje wprowadzaną wartość, a  $N$  jest wielkością tablicy haszującej.
  - (a) Użytkownik powinien mieć możliwość zdefiniowania ilości wprowadzanych elementów (należy zwerifikować, że jest to liczba pierwsza),
  - (b) wartości zapisywane do tablicy powinny być generowane losowo bez powtórzeń,
  - (c) należy zaimplementować trzy sposoby rozwiązywania kolizji:
    - i. linkowanie
    - ii. próbkowanie liniowe
    - iii. podwójne haszowanie z drugą funkcją haszującą w postaci  $q - x\%q$ , gdzie  $q < N$  jest liczbą pierwszą.
  - (d) należy zaprezentować działanie operacji dodawania, wyszukiwania oraz usuwania elementów z tablicy, za każdym razem wyświetlając ilość próbek potrzebnych do dodania i wyszukania elementów.
2. Należy zaimplementować graf przechowujący elementy określonego typu. Należy napisać funkcje wykonujące podstawowe operacje na grafie zgodnie z informacjami podawanymi na wykładzie.
  - (a) Należy zaimplementować graf za pomocą listy sąsiedztwa zgodnie z wytycznymi na wykładzie
  - (b) Należy zaimplementować graf za pomocą macierzy sąsiedztwa zgodnie z wytycznymi na wykładzie.

### **Zadania na ocenę db (4.0):**

1. Należy zaimplementować graf przechowujący elementy określonego typu. Należy napisać funkcje wykonujące podstawowe operacje na grafie zgodnie z informacjami podawanymi na wykładzie.
  - (a) Należy zaimplementować graf za pomocą listy sąsiedztwa zgodnie z wytycznymi na wykładzie
  - (b) Należy zaimplementować graf za pomocą macierzy sąsiedztwa zgodnie z wytycznymi na wykładzie.
2. Należy zaimplementować algorytmy Kruskala i Prima oraz przeprowadzić analizę efektywności tych algorytmów zgodnie z informacjami w Uwadze 3.

### **Zadania na ocenę bdb (5.0) - do wyboru pkt. 1 lub 2.:**

1. Należy zaimplementować grę w kółko i krzyżyk z wykorzystaniem algorytmu MinMax z alfa-beta cięciami. Gracz powinien posiadać możliwość definiowania rozmiaru pola (kwadratowego) wraz z ilością znaków w rzędzie.
2. Należy zaimplementować grę w warcaby z wykorzystaniem algorytmu MinMax z alfa-beta cięciami. Należy doprecyzować zasady (np. czy bicie jest obowiązkowe, etc.)
3. Rozwiązania bez zastosowania alfa-beta cięć będą ocenione maksymalnie na db+ (4.5).

## **Sprawozdanie**

Sprawozdanie powinno zawierać:

- krótkie wprowadzenie,
- opis badanych algorytmów z omówieniem ich złożoności obliczeniowej (w przypadku gry: opis tworzonej gry wraz z wytłumaczeniem stosowanych technik SI)
- omówienie przebiegu eksperymentów i przedstawienie uzyskanych wyników (w postaci tabel i wykresów) - nie dotyczy gry,
- podsumowanie i wnioski (w przypadku niezgodności uzyskanych wyników z przewidywanymi spróbować wyjaśnić przyczyny),
- literatura (materiały wykorzystane do wykonania ćwiczenia, w tym strony internetowe).

## **Bibliografia**

1. Cormen T., Leiserson C.E., Rivest R.L., Stein C., Wprowadzenie do algorytmów, WNT
2. Drozdek A., C++. Algorytmy i struktury danych, Helion
3. <http://www.algorytm.org/algorytmy-grafowe/algorytm-prima.html>
4. <http://www.algorytm.org/algorytmy-grafowe/algorytm-kruskala.html>
5. <http://users.v-lo.krakow.pl/~toma/algorytmy/Algorytmy%20grafowe.pdf>