

# **Tematy projektów z Algorytmów geometrycznych**

## **Rok akademicki 2024/2025**

### **1. Lokalizacja punktu w przestrzeni dwuwymiarowej – metoda separatorów**

Dany jest obszar z podziałem poligonowym. Zadawany jest punkt P na płaszczyźnie. Należy zaimplementować algorytm lokalizacji punktu metodą separatorów, który odpowie na pytanie, w którym elemencie znajduje się dany punkt. (Należy także umożliwić użytkownikowi odpowiednio zadawanie obszaru z podziałem poligonowym.) Zrobić analizę efektywności algorytmu.

Program powinien w sposób graficzny prezentować etapy algorytmu dla wybranych przykładów (w celu objaśnienia działania algorytmu). Program ma służyć jako narzędzie dydaktyczne do objaśnienia działania algorytmu.

---

### **2. Lokalizacja punktu w przestrzeni dwuwymiarowej – metoda trapezowa**

Dany jest obszar z podziałem poligonowym. Zadawany jest punkt P na płaszczyźnie. Należy zaimplementować algorytm lokalizacji punktu metodą trapezową, który odpowie na pytanie, w którym elemencie znajduje się dany punkt. (Należy także umożliwić użytkownikowi odpowiednio zadawanie obszaru z podziałem poligonowym.) Zrobić analizę efektywności algorytmu.

Program powinien w sposób graficzny prezentować etapy algorytmu dla wybranych przykładów (w celu objaśnienia działania algorytmu). Program ma służyć jako narzędzie dydaktyczne do objaśnienia działania algorytmu.

---

### **3. Lokalizacja punktu w przestrzeni dwuwymiarowej – metoda doskonalenia triangulacji – algorytm Kirkpatrick’a**

Dany jest obszar z podziałem poligonowym. Zadawany jest punkt P na płaszczyźnie. Należy zaimplementować algorytm Kirkpatrick’a lokalizacji punktu, który odpowie na pytanie, w którym elemencie znajduje się dany punkt. (Należy także umożliwić użytkownikowi odpowiednio zadawanie obszaru z podziałem poligonowym.) Zrobić analizę efektywności algorytmu.

Program powinien w sposób graficzny prezentować etapy algorytmu dla wybranych przykładów (w celu objaśnienia działania algorytmu). Program ma służyć jako narzędzie dydaktyczne do objaśnienia działania algorytmu.

---

#### **4. Otoczka wypukła dla zbioru punktów w przestrzeni dwuwymiarowej**

Należy zaimplementować omawiane na wykładzie algorytmy wyznaczania otoczki wypukłej dla zbioru punktów w przestrzeni dwuwymiarowej: przyrostowy, górna i dolna otoczka, Quickhull, dziel i rządz oraz Chan'a. Porównać je między sobą oraz z algorytmami zaimplementowanymi na ćwiczeniach (Grahama i Jarvisa). Dobrać odpowiednio zbiory testowe. Porównać efektywność algorytmów.

Program powinien w sposób graficzny prezentować etapy konstrukcji otoczki (w celu objaśnienia działania algorytmu) dla każdego z algorytmów. Program ma służyć jako narzędzie dydaktyczne do objaśnienia działania algorytmów.

---

#### **5. Wieloboki Voronoi – porównanie metod konstrukcji**

Należy zaimplementować dwa różne algorytmy wyznaczania wieloboków Voronoi dla zadanego zbioru punktów na płaszczyźnie (wykorzystując metrykę euklidesową).

Przetestować te algorytmy na różnych zbiorach danych i przeanalizować różnice w obu podejściach.

Program powinien w sposób graficzny prezentować etapy algorytmu dla wybranych przykładów (w celu objaśnienia działania algorytmu).

---

#### **6. Triangulacja wielokąta prostego – porównanie metod**

Należy zaimplementować algorytmy konstrukcji triangulacji dla dowolnego wielokąta prostego dwoma metodami. W obu przypadkach triangulacja ma się opierać jedynie na wierzchołkach wielokąta (bez dodawania wierzchołków wewnątrz wielokąta). Pierwsza metoda to podział wielokąta na wielokąty monotoniczne (zgodnie z algorytmem podanym na wykładzie), a następnie triangulacja wynikłych z podziału wielokątów monotonicznych także metodą podaną na wykładzie (i realizowaną na ćwiczeniach). Drugi algorytm to triangulacja Delaunay'a z ograniczeniami (z procedurami odzyskania krawędzi i usunięcia trójkątów zewnętrznych). W tej metodzie wybrać odpowiedni sposób poszukiwania trójkąta w istniejącej triangulacji (i go omówić). Program powinien pozwolić na wizualizację działania obu metod.

Przeanalizować i porównać obie metody pod kątem efektywności oraz jakości otrzymanej triangulacji (wybrać odpowiednie kryterium oceny trójkątów).

---

#### **7. Wyszukiwanie geometryczne – przeszukiwanie obszarów ortogonalnych – quadtree oraz kd-drzewa**

Dane – zbiór punktów  $P$  na płaszczyźnie. Zapytanie: dla zadanych  $x_1, x_2, y_1, y_2$  znaleźć punkty  $q$  ze zbioru  $P$  takie, że  $x_1 \leq q_x \leq x_2, y_1 \leq q_y \leq y_2$ .

Celem projektu jest zaimplementowanie odpowiednich struktur danych – quadtree oraz kd-drzew, które pozwalają szybko odpowiadać na takie zapytania.

Program ma służyć jako narzędzie dydaktyczne do objaśnienia tworzenia struktury i realizacji zapytań. Projekt dwuosobowy – każda z osób implementuje inną strukturę i razem porównują oba podejścia. Istotne są analiza i porównanie algorytmów!

---

## 8. Przetwarzanie i przechowywanie opisu siatki trójkątnej na płaszczyźnie

Program powinien najpierw wczytać zawartość przykładowych triangulacji 2D, zapisanych w postaci tekstowej jako lista punktów (dwie współrzędne rzeczywiste dla każdego punktu) oraz lista elementów (każdy trójkąt opisany przez numery trzech wierzchołków, punkty numerowane są od zera), a następnie zilustrować wczytaną siatkę. Kolejno program powinien tworzyć odpowiednie struktury danych i odtwarzać brakujące obiekty/powiązania tak, aby otrzymać reprezentację „Half Edge Data Structure”.

Porównać pod kątem kosztu pamięciowego i czasowego obie reprezentacje (bezpośrednio pobraną z pliku – lista wierzchołków i tablica połączeń, oraz Half Edge Data Structure) przy przeprowadzeniu następujących operacji (każdej z osobna):

- OP1: wyznaczanie otoczenia dla wybranego wierzchołka (kolejne warstwy incydentnych wierzchołków – należy rozpatrzyć otoczenia składające się z jednej warstwy oraz dwóch warstw),
- OP2: wyznaczanie otoczenia dla wybranego trójkąta (kolejne warstwy incydentnych trójkątów – należy rozpatrzyć otoczenia składające się z jednej warstwy oraz dwóch warstw),
- OP3: przeglądanie incydentnych trójkątów od wybranego trójkąta w kierunku wybranego punktu (dla odszukania trójkąta zawierającego dany punkt).

Program powinien mieć możliwość prezentowania w sposób graficzny etapów algorytmu dla wybranych przykładów (w celu objaśnienia działania algorytmu).

---

## 9. Wyznaczanie minimalnego okręgu i prostokąta zawierającego chmurę punktów na płaszczyźnie

Zadawana jest chmura punktów na płaszczyźnie dwuwymiarowej. Program ma wyznaczać:

- minimalny okrąg zawierający tę chmurę,
- prostokąt o minimalnym polu powierzchni zawierający tę chmurę.
- prostokąt o minimalnym obwodzie zawierający tę chmurę.

Program powinien w sposób graficzny prezentować etapy algorytmu dla wybranych przykładów (w celu objaśnienia działania algorytmu). *Projekt 1-osobowy – tylko dla grupy z nieparzystą liczbą studentów*

---