Geometria obliczeniowa

Zestaw 2 – Triangulacja wielokąta

Napisać program wyznaczający triangulację wielokąta prostego. Program ma mieć następujące możliwości:

1. wczytywanie wierzchołków wielokąta z pliku tekstowego (w pierwszej linii pliku podana jest liczba wierzchołków n, a w kolejnych n liniach współrzędne wierzchołków); wierzchołki są w porządku przeciwnym do ruchu wskazówek zegara,

2. podawanie wielokąta za pomocą myszki; w przypadku gdy użytkownik poda wierzchołki zgodnie z ruchem wskazówek zegara należy dokonać konwersji listy wierzchołków do porządku przeciwnego do ruchu wskazówek zegara,

3. rysowania wielokąta wraz z wyznaczoną triangulacją.

Pierwszym krokiem jaki należy wykonać przy wyznaczaniu triangulacji jest podział wielokąta prostego na części monotoniczne. Do tego celu należy skorzystać z algorytmu 1, w którym do klasyfikacji wierzchołka można skorzystać z algorytmu 2. Do triangulacji wielokąta monotonicznego należy wykorzystać algorytm 3.

Literatura

[1] de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Geometria obliczeniowa algorytmy i zastosowania. WNT, Warszawa, (2007)

[2] Garey, M.R., Johnson, D.S., Preparata, F.P., Tarjan, R.E.: Triangulating a Simple Polygon. Information Processing Letters 7(4), 175-179, (1978)

[3] Lee, D.T., Preparata, F.P.: Location of a Point in a Planar Subdivision and its Applications. SIAM Journal of Computing 6(3), 594-606, (1977)

Algorytm 1: Podział wielokąta prostego na wielokąty monotoniczne

Dane: Wielokąt prosty P pamiętany w podwójnie łączonej liście krawędzi $\mathcal{D}$.

Wynik: Podział P na wielokąty monotoniczne pamiętany w $\mathcal{D}$.

MakeMonotone(P)
- Dla wierzchołków z P stwórz kolejkę priorytetową $\mathcal{Q}$ używającą jako priorytetu ich współrzędnych y. Jeśli dwa punkty mają taką samą współrzędną y, to wyższy priorytet ma ten z mniejszą współrzędną x.
- Inicjuj puste drzewo wyszukiwań binarnych $\mathcal{T}$.
- **while** *$\mathcal{Q}$ nie jest puste* **do**
  - Usuń z $\mathcal{Q}$ wierzchołek v_i o najwyższym priorytecie.
  - Wywołaj odpowiednią procedurę badającą v_i zależnie od jego rodzaju.

HandleStartVertex(v_i)
- Wstaw e_i do $\mathcal{T}$ i ustaw $helper(e_i)$ na v_i.

HandleEndVertex(v_i)
- **if** *$helper(e_{i-1})$ jest wierzchołkiem łączącym* **then**
  - Wstaw do $\mathcal{D}$ przekątną łączącą v_i z $helper(e_{i-1})$.
- Usuń e_{i-1} z $\mathcal{T}$.

HandleSplitVertex(v_i)
- Szukaj w $\mathcal{T}$ krawędzi e_j bezpośrednio na lewo od v_i.
- Wstaw do $\mathcal{D}$ przekątną łączącą v_i z $helper(e_j)$.
- $helper(e_j) = v_i$
- Wstaw e_i do $\mathcal{T}$ i ustaw $helper(e_i)$ na v_i.

HandleMergeVertex(v_i)
- **if** *$helper(e_{i-1})$ jest wierzchołkiem łączącym* **then**
  - Wstaw do $\mathcal{D}$ przekątną łączącą v_i z $helper(e_{i-1})$.
- Usuń e_{i-1} z $\mathcal{T}$.
- Szukaj w $\mathcal{T}$ krawędzi e_j bezpośrednio na lewo od v_i.
- **if** *$helper(e_j)$ jest wierzchołkiem łączącym* **then**
  - Wstaw do $\mathcal{D}$ przekątną łączącą v_i z $helper(e_j)$.
- $helper(e_i) = v_i$

HandleRegularVertex(v_i)
- **if** *wnętrze P leży na prawo od v_i* **then**
  - **if** *$helper(e_{i-1})$ jest wierzchołkiem łączącym* **then**
    - Wstaw do $\mathcal{D}$ przekątną łączącą v_i z $helper(e_{i-1})$.
  - Usuń e_{i-1} z $\mathcal{T}$.
  - Wstaw e_i do $\mathcal{T}$ i ustaw $helper(e_i)$ na v_i.
- **else**
  - Szukaj w $\mathcal{T}$ krawędzi e_j bezpośrednio na lewo od v_i.
  - **if** *$helper(e_j)$ jest wierzchołkiem łączącym* **then**
    - Wstaw do $\mathcal{D}$ przekątną łączącą v_i z $helper(e_j)$.
  - $helper(e_j) = v_i$

Algorytm 2: Klasyfikacja rodzaju wierzchołka

Dane: v – wierzchołek do sklasyfikowania
Wynik: Rodzaj wierzchołka

p = poprzedni wierzchołek v
n = następny wierzchołek v
if $v_y > p_y \wedge v_y > n_y$ **then**
    if *p, v, n tworzą skręt w lewo* **then**
        return Start
    else
        return Split
else
    if $v_y < p_y \wedge v_y < n_y$ **then**
        if *p, v, n tworzą skręt w lewo* **then**
            return End
        else
            return Merge
return Regular

Algorytm 3: Triangulacja wielokąta ściśle y-monotonicznego

Dane: Ściśle y-monotoniczny wielokąt P pamiętany w podwójnie łączonej liście krawędzi $\mathcal{D}$.
Wynik: Triangulacja wielokąta P pamiętana w podwójnie łączonej liście krawędzi $\mathcal{D}$.

Połącz wierzchołki z lewego łańcucha i wierzchołki z prawego łańcucha wielokąta P w jeden ciąg, uporządkowany względem współrzędnej y. Jeśli dwa wierzchołki mają taką samą współrzędną y, to pierwszy jest ten bardziej na lewo. Niech $u_1, u_2, \ldots, u_n$ oznacza uporządkowany ciąg.
Inicjuj pusty stos $\mathcal{S}$ i włóż na niego u_1 i u_2.
for $j = 3$ **to** $n - 1$ **do**
    if *u_j i wierzchołek na szczycie stosu $\mathcal{S}$ są na różnych łańcuchach* **then**
        Zdejmij wszystkie wierzchołki z $\mathcal{S}$.
        Wstaw do $\mathcal{D}$ przekątne z u_j do każdego ze zdjętych wierzchołków, oprócz ostatniego.
        Włóż u_{j-1} i u_j na $\mathcal{S}$.
    else
        Zdejmij jeden wierzchołek z $\mathcal{S}$.
        Zdejmuj pozostałe wierzchołki z $\mathcal{S}$ tak długo, jak długo przekątne z u_j do nich są wewnątrz P. Wstaw te przekątne do $\mathcal{D}$. Włóż ostatni wierzchołek, który został zdjęty z powrotem na $\mathcal{S}$.
        Włóż u_j na $\mathcal{S}$.
Dodaj przekątne z u_n do wszystkich wierzchołków na stosie, oprócz pierwszego i ostatniego.