

Dokumentacja techniczna projektu „Wyznaczanie minimalnego okręgu i prostokąta zawierającego chmurę punktów 2D”

randPoints(num,range) O(num)

num-liczba punktów

range-zakres liczbowy

Funkcja przyjmuje na wejście parę liczb typu int i zwraca zbiór losowo generowanych punktów w postaci listy elementów (float,float) w zakresie liczb rzeczywistych [-range, +range].

det(point1,point2,point3,which) O(1)

point1,point2,point3 -punkty 2D zadane w postaci (float,float)

which-przełącznik sposobu liczenia wyznacznika, 0 wyznacznik 3x3,1 wyznacznik 2x2

Funkcja zwraca wyznacznik liczony w kolejności point1->point2->point3 w postaci float.

findStart(points) O(points)

points -punkty 2D zadane w postaci listy elementów (float,float)

Funkcja zwraca minimum zadanej listy punktów względem 2 współrzędnej oraz usuwa minimalny punkt z listy.

powDistance(a,b) O(1)

a,b -punkty 2D zadane w postaci (float,float)

Funkcja zwraca kwadrat odległości między punktami.

jarvis(points,precision,filename) O(points*k) ,gdzie k-liczba punktów otoczki

points -punkty 2D zadane w postaci listy elementów (float,float)

precision-precyzja wyznaczania punktów współliniowych zadana w formie liczby wymiernej o dolnym ograniczeniu 10^{-11}

filename(opcjonalny)-nazwa pliku do którego zostanie zapisana otoczka w formie listy punktów

Funkcja zwraca otoczkę wypukłą zadanej listy punktów w postaci listy punktów (float,float) w kolejności przeciwnej do ruchu wskazówek zegara.

splitConvexHullByX(hull) O(hull)

hull-otoczka wypukła zadana w postaci listy punktów (float,float) w kolejności przeciwnej do ruchów wskazówek zegara

Funkcja zwraca listy u oraz l w postaci (u,l), zawierające odpowiednio górny oraz dolny łańcuch otoczki w postaci listy punktów (float,float) posortowanych względem współrzędnej x układu współrzędnych.

getAntiPodal (u,l) O(u+l)

u-górny łańcuch otoczki zadany w postaci listy punktów (float float) posortowanych względem współrzędnej x

l- dolny łańcuch otoczki zadany w postaci listy punktów (float float) posortowanych względem współrzędnej x

Funkcja zwraca listę par punktów w postaci ((float,float),(float,float)) ,które są antypodyczne dla danego wielokąta, w kolejności zgodnej z ruchem wskazówek zegara.

getMiddlePoint(point1,point2) O(1)

point1,point2-punkty zadane w postaci (float,float)

Funkcja zwraca punkt w postaci (float,float) będący punktem środkowym na odcinku pomiędzy point1 i point2.

getCircleFromPoints(point1,point2,point3) O(1)

point1,point2,point3-punkty zadane w postaci (float,float)

Funkcja zwraca okrąg opisany na punktach point1,point2,point3 w postaci punktu środka oraz promienia zwracanych jako ((float,float),float).

isEnclosing(hull,middle,radius) O(hull)

hull-otoczka wypukła zadana w postaci listy punktów (float,float) w kolejności przeciwnej do ruchów wskazówek zegara

middle-punkt środka zadany w postaci (float,float)

radius-promień zadany w postaci float

Funkcja zwraca true jeśli wszystkie punkty zawarte w hull znajdują się wewnątrz okręgu lub na okręgu, w przeciwnym wypadku zwraca false.

calculateMinimalCircle(podal,hull) O(hull)

podal- lista par punktów w postaci ((float,float),(float,float)) ,które są antypodyczne dla danego wielokąta, w kolejności zgodnej z ruchem wskazówek zegara

hull-otoczka wypukła zadana w postaci listy punktów (float,float) w kolejności przeciwnej do ruchów wskazówek zegara

Funkcja zwraca minimalny okrąg zawierający w postaci (((float,float),float,float,int), [(float,float)]), kolejno:

Punkt środka, promień, kwadrat odległości do prostej dzielącej, numer iteracji, w której znaleziono okrąg, lista punktów wszystkich rozpatrywanych okręgów (w celu wizualizacji)

getCircle(point,radius,num) O(num)

point- punkt środka okręgu w postaci (float,float)

radius- promień okręgu w postaci float

num- liczba punktów w postaci int

Funkcja zwraca listę punktów w postaci (float,float) leżących na zadanym okręgu.

vecLength(vec) O(1)

vec- wektor zadany w postaci [float,float]

Funkcja zwraca długość wektora w postaci float.

getPerpVec(vec) O(1)

vec- wektor zadany w postaci [float,float]

Funkcja zwraca wektor prostopadły [y,-x] do zadanego w postaci [float,float]

getMinAngle(vec1,vec2) O(vec1)

vec1,vec2- listy wektorów zadane w postaci [float,float]

Funkcja porównuje kąty jakie tworzą kolejne pary wektorów [vec1[i],vec2[i]] między sobą i zwraca numer iteracji w której został znaleziony minimalny kąt powiększony o 1.

calculateBoundingBox(hull) O(hull)

hull-otoczka wypukła zadana w postaci listy punktów (float,float) w kolejności przeciwnej do ruchów wskazówek zegara

Funkcja zwraca dane w postaci (((float,float),(float,float),(float,float),(float,float)),
((float,float),(float,float),(float,float),(float,float))),[LinesCollection])

Kolejno:

Prostokąt o minimalnym polu zadany jako punkty wierzchołków ((float,float),(float,float),
(float,float),(float,float))

Prostokąt o minimalnym obwodzie zadany jako punkty wierzchołków

((float,float),(float,float),(float,float),(float,float))

Lista kolekcji zawierająca wszystkie prostokąty zawierające stworzone podczas pracy algorytmu (dla wizualizacji działania)

Klasy:

supportLine

Klasa składa się z wektora oraz punktu i opisuje prostą wsparcia, posiada jedynie konstruktor przyjmujący na wejście wektor w postaci [float,float] oraz punkt wsparcia w postaci (float,float)