Tutorial-2 Solution

Answer to Q no. 1(a)

Determining Upper Tangent:

T_1_index = get_rightmostpoint_index $(P, n) = 6$

T_2_index = get_leftmostpoint_index $(Q, m) = 9$

$T_1 = P_6$

$T_2 = Q_9$

done $= 0$

Iteration-1:

done $= 1$

~~Subiteration-1:~~ Traversing convex hull Q

Iteration 1.1:

$i = 9 \quad T_2 = Q_9$

$CCW(P_6, Q_9, Q_0) > 0$

Iteration 1.2:

$i \% m = 0 \quad T_2 = Q_0$

$CCW(P_6, Q_0, Q_1) > 0$

~~if~~

iv) if (CCW(points[i], points[i+1], Q) == 0) return 2

v) else if (CCW(points[i], points[i+1], Q) > 0 &&
CCW(points[0], points[n-1], Q) == 0) return 2

vi) else if (CCW(points[i], points[i+1], Q) > 0 &&
CCW(points[0], points[1], Q == 0) return 2

vii) else if (CCW(points[i], points[i+1], Q) > 0) return 1.

viii) else return 0.

point_in_polygon_query (P[], N, Q[], M)

i) calculate P_0_index = get_leftmostpoint_index(P, N).

ii) rotate (P.begin(), P.begin() + P_0_index, P.end()).

iii) for i = 0 to Q-1:

iii. a. calculate f = point_in_polygon(P, N, Q[i]).

iii. b. if (f == 1) print "Point is inside polygon".
else if (f == 2) print "Point on boundary of polygon".
else print "Point is outside polygon"

Tutorial-2 Solution

Answer to Q no. 1(a)

Determining Upper Tangent:

T_1_index = get_rightmostpoint_index $(P, n) = 6$

T_2_index = get_leftmostpoint_index $(Q, m) = 9$

$T_1 = P_6$

$T_2 = Q_9$

done $= 0$

Iteration-1:

done $= 1$

~~Subiteration-1:~~ Traversing convex hull Q

Iteration 1.1:

$i = 9 \quad T_2 = Q_9$

$CCW(P_6, Q_9, Q_0) > 0$

Iteration 1.2:

$i \% m = 0 \quad T_2 = Q_0$

$CCW(P_6, Q_0, Q_1) > 0$

Iteration-1.3:

$i\%m = 1 \quad T_2 = Q_1$

$CCW(P_6, Q_1, Q_2) < 0$

$\because i\%m = 1 != 9$

$\therefore$ done $= 0$

$T_2 = Q_1 \quad T_2\text{-index} = 1$

Traversing convex hull P

Iteration-1.1:

$(j+n)\%n = 6 \quad T_1 = P_6$

$CCW(Q_1, P_6, P_5) < 0$

Iteration-1.2:

$(j+n)\%n = 5 \quad T_1 = P_5$

$CCW(Q_1, P_5, P_4) < 0$

Iteration-1.3:

$(j+n)\%n = 4 \quad T_1 = P_4$

$CCW(Q_1, P_4, P_2) > 0$

$\because (j+n)\%n = 4 != 6$

$\therefore$ done $= 0$.

$T_1 = P_4$ T_1- index $= 4$

Iteration-2:

done $= 1$

Convex hull Q

Iteration-2.1:

$i\%m = 1$ $T_2 = Q_1$

$ccw(P_4, Q_1, Q_2) > 0$

Iteration-2.2:

$i\%m = 2$ $T_2 = Q_2$

$ccw(P_4, Q_2, Q_3) < 0$

$\because$ $i\%m = 2 \,!= 1$

$\therefore$ done $= 0$

$T_2 = Q_2$ and T_2-index $= 2$

Convex hull P

Iteration-2.1:

$(j+n)\%n = 4$ $T_1 = 4$

$ccw(Q_2, P_4, P_2) > 0$

$T_1 = P_4$ and T_1-index $= 4$

Iteration-3:

done = 1

Convex hull Q

Iteration-3.1:

$i\%m = 2 \quad T_2 = Q_2$

$\cancel{CCW(Q_2, P_4, P_2)}$

$CCW(P_4, Q_2, Q_3) < 0$

$T_2 = Q_2$ and T_2-index $= 2$

Convex hull P

Iteration-3.1:

$(j+n)\%n = 4 \quad T_1 = P_4$

$CCW(Q_2, P_4, P_2) > 0$

$T_1 = P_4$ and T_1-index $= 4$

So, the upper tangent contains the points P_4 and Q_2.

Determining Lower Tangent:

T_1-index = get_rightmostpoint_index $(P, n) = 6$

T_2-index = get_leftmostpoint_index $(Q, m) = 9$

$T_1 = P_6$

$T_2 = Q_9$

done = 0

Iteration-1:

done = 1

For ~~polyg~~ convex hull Q:

Iteration-1.1:

~~i%m = 9~~ $(i+m)\%m = 9$ $T_2 = Q_9$

$CCW(P_6, Q_9, Q_8) < 0$

Iteration-1.2:

$(i+m)\%m = 8$ $T_2 = Q_8$

$CCW(P_6, Q_8, Q_7) < 0$

Iteration-1.3:

$(i+m)\%m = 7 \quad T_2 = Q_7$

$CCW(P_6, Q_7, Q_6) > 0$

$\because (i+m)\%m \,!= 9$

$\therefore done = 0$

$T_2 = Q_7$ and T_2-index $= 7$

Traversing convex hull P:

Iteration-1.1:

$j\%n = 6 \quad T_1 = P_6$

$CCW(Q_7, P_6, P_0) > 0$

Iteration-1.2:

$j\%n = 0 \quad T_1 = P_0$

$CCW(Q_7, P_0, P_1) < 0$

$\because j\%n \,!= 6$

$\therefore done = 0$

$T_1 = P_0$ and T_1-index $= 0$

Iteration-2:

done=1

Traversing convex hull Q:

Iteration-2.1

$(i+m)\%m = 7$ $T_2 = Q_7$

$CCW(P_0, Q_7, Q_6) > 0$

$T_2 = Q_7$ and T_2-index = 7

Traversing convex hull P:

Iteration-2.1

$(j\%n) = 0$ $T_1 = P_0$

$CCW(Q_7, P_0, P_1) < 0$

$T_1 = P_0$ and T_1-index = 0

So, the lower tangent contains the points P_0 and P_2

∴ The merged convex hull,

$CH[\,] = \{P_0, P_1, P_2, P_4, Q_2, Q_3, Q_4, Q_5, Q_6\}$

Answer to Qno.1(b)

The given algorithm is not correct as the vertices of the polygon are in clockwise order and the algorithm will return false for a right turn, i.e., clockwise orientation.

The correct algorithm is as follows:

bool is_convex($(P_0, \ldots, P_{n-1})$ {
  for $i = 0, \ldots, n-1$:
    ~~if (P_i, $P_{(i+1) \bmod n}$~~
    if (CCW($P_i, P_{(i+1)\%n}, P_{(i+2)\%n}$) > 0); ~~ret~~
        return false;

  return true;
}

Answer to Q no. 1 (c)

This algorithm takes a convex polygon and a query of point as input and returns for each query if the point is inside, outside or on the boundary of the polygon.

crossproduct(P_1, P_2, P_3)

i) calculate $\overrightarrow{P_1P_2} = P_2 - P_1 = (P_2.x - P_1.x, P_2.y - P_1.y)$

ii) calculate $\overrightarrow{P_1P_3} = P_3 - P_1 = (P_3.x - P_1.x, P_3.y - P_1.y)$

iii) calculate $\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} = \{(P_2.x - P_1.x)(P_3.y - P_1.y) - (P_3.x - P_1.x)(P_2.y - P_1.y)\}$

iv) return $\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}$.

CCW(P_1, P_2, P_3)

i) calculate CP = crossproduct(P_1, P_2, P_3)

ii) return CP.

get_leftmostpoint_index (points[], n)

i) set min_x = inf and min_ind = -1.

ii) for i=0 to n-1:

ii.a. if(points[i].x < min_x || (points[i].x == min_x && points[i].y > points[min_ind].y))

set min_x = points[i].x and min_ind = i.

iii) return min_ind.

point_in_polygon(points[], n, Q)

i) if (ccw(points[0], points[1], Q) < 0 || ccw(points[0], points[n-1], Q) > 0) return 0.

ii) set start = 0 and end = n-2.

iii) while (start <= end):

iii.a. calculate mid = (start+end)/2.

iii.b. set i = mid

iii.c. if(ccw(points[0], points[i], Q) >= 0) set start = mid+1.

else set end = mid-1.

~~if~~

iv) if (CCW(points[i], points[i+1], Q) == 0) return 2

v) else if (CCW(points[i], points[i+1], Q) > 0 &&
CCW(points[0], points[n-1], Q) == 0) return 2

vi) else if (CCW(points[i], points[i+1], Q) > 0 &&
CCW(points[0], points[1], Q == 0) return 2

vii) else if (CCW(points[i], points[i+1], Q) > 0) return 1.

viii) else return 0.

point_in_polygon_query (P[], N, Q[], M)

i) calculate P_0_index = get_leftmostpoint_index(P, N).

ii) rotate (P.begin(), P.begin() + P_0_index, P.end()).

iii) for i = 0 to Q-1:

iii. a. calculate f = point_in_polygon(P, N, Q[i]).

iii. b. if (f == 1) print "Point is inside polygon".
else if (f == 2) print "Point on boundary of polygon".
else print "Point is outside polygon".

Answers to Qno.1(d)

Step-1:

A = get_area_of_simple_polygon($V_0(7,1), V_1(13,2), V_2(14,7), V_3(10,9), V_4(6,10), V_5(5,9), V_6(4,6), V_7(4,3)$)

$\overrightarrow{OV_0} \times \overrightarrow{OV_1}$ = crossproduct$((0,0),(7,1),(13,2))$
= crossproduct$((7,1),(13,2))$
$= \begin{vmatrix} 7 & 13 \\ 1 & 2 \end{vmatrix} = (14-13) = 1$

$\overrightarrow{OV_1} \times \overrightarrow{OV_2}$ = crossproduct$((13,2),(14,7)) = 63$

$\overrightarrow{OV_2} \times \overrightarrow{OV_3}$ = crossproduct$((0,0),(14,7),(10,9)) = 56$

$\overrightarrow{OV_3} \times \overrightarrow{OV_4}$ = crossproduct$((0,0),(10,9),(6,10)) = 46$

$\overrightarrow{OV_4} \times \overrightarrow{OV_5}$ = crossproduct$((0,0),(6,10),(5,9)) = 4$

$\overrightarrow{OV_5} \times \overrightarrow{OV_6}$ = crossproduct$((0,0),(5,9),(4,6)) = -6$

$\overrightarrow{OV_6} \times \overrightarrow{OV_7}$ = crossproduct$((0,0),(4,6),(4,3)) = -12$

$\overrightarrow{OV_7} \times \overrightarrow{OV_0}$ = crossproduct$((0,0),(4,3),(7,1)) = -17$

$\therefore A = |1+63+56+46+4-6-12-17| / 2.0 = 67.5$

<u>Step-2:</u>

$B = 8$

<u>Step-3:</u>

Iteration-0:

$B = B + \text{get\_LatticePoint\_onLS}((7,1),(13,2))$

$= 8 + \gcd(|2-1|, |13-7|) - 1$

$= 8 + 1 - 1$

$= 8$

Iteration-1:

$B = B + \text{get\_LatticePoint\_onLS}((13,2),(14,7))$

$= 8 + \gcd(|7-2|, |14-13|) - 1$

$= 8 + 1 - 1$

$= 8$

Iteration-2:

$B = B + \text{get\_LatticePoint\_onLS}((14,7),(10,9))$

$= 8 + \gcd(|9-7|, |10-14|) - 1$

$= 8 + 2 - 1$

$= 9$

Iteration-3:

$B = B + \text{get\_LatticePoint\_onLS}((10,9),(6,10))$

$= 9 + \gcd(|10-9|, |6-10|) - 1$

$= 9 + 1 - 1$

$= 9$

Iteration-4:

$B = B + \text{get\_LatticePoint\_onLS}((6,10),(5,9))$

$= 9 + \gcd(|9-10|, |5-6|) - 1$

$= 9 + 1 - 1$

$= 9$

Iteration-5:

$B = B + \text{get\_LatticePoint\_onLS}((5,9),(4,6))$

$= 9 + \cancel{\text{get}}\ \gcd(|6-9|, |4-5|) - 1$

$= 9 + 1 - 1$

$= 9$

Iteration-6:

$B = B + \text{get\_LatticePoint\_onLS}((4,6),(4,3))$

$= 9 + \gcd(|3-6|, |4-4|) - 1$

$= 9 + 3 - 1$

$= 11$

Iteration-7:

$B = B + \text{get\_LatticePoint\_onLS}((4,3),(7,1))$

$= 11 + \gcd(|1-3|, |7-4|) - 1$

$= 11 + 1 - 1$

$= 11$

Step-4:

$I = A - \frac{B}{2} + 1$

$= 67.5 - \frac{11}{2} + 1$

$= 63$