Algoritmi avansaţi

Laborator 5 (săpt. 9 şi 10)

1. (0,5p) *Implementaţi / utilizaţi testul de orientare.*

Input. Trei puncte $P = (x_P, y_P), Q = (x_Q, y_Q), R = (x_R, y_R)$ (în această ordine) din $\mathbb{R}^2$.
Output. Programul afişează natura virajului PQR (viraj la stânga, viraj la dreapta, puncte coliniare).

2. (0,5p) *Algoritm cu complexitate-timp liniară pentru frontiera acoperirii convexe a unui poligon dat.*

Input. Numărul de vârfuri n, vârfurile poligonului: $P_1 = (x_{P_1}, y_{P_1}), P_2 = (x_{P_2}, y_{P_2}), \ldots, P_n = (x_{P_n}, y_{P_n})$ (în această ordine) din $\mathbb{R}^2$.
Output. Programul afişează vârfurile acoperirii convexe a mulţimii $\{P_1, \ldots, P_n\}$.
Precizare. Pentru testare, $P_1P_2 \ldots P_n$ reprezintă un poligon parcurs în sens trigonometric (acest lucru nu mai trebuie verificat). Algoritmul va avea complexitatea-timp liniară.

3. (1p) *Algoritm eficient pentru stabilirea poziţiei unui punct faţă de un poligon convex.*

Input. Numărul de vârfuri n, vârfurile poligonului convex $P_1 = (x_{P_1}, y_{P_1}), P_2 = (x_{P_2}, y_{P_2}), \ldots, P_n = (x_{P_n}, y_{P_n})$ (în această ordine), un punct Q din $\mathbb{R}^2$.
Output. Programul afişează poziţia relativă a punctului Q faţă de poligon (în interior, în exterior, pe laturi).
Precizare. Pentru testare, $P_1P_2 \ldots P_n$ reprezintă un poligon convex parcurs în sens trigonometric (acest lucru nu mai trebuie verificat). Algoritmul va fi cât mai eficient.

4. (1p) *Implementaţi* algoritmul *care construieşte, în context euclidian, un traseu optim pentru TSP folosind acoperirea convexă.*

Algoritmi avansaţi

Laborator 6 (săpt. 11 şi 12)

1. (1p) *Poziţia unui punct faţă de un poligon - algoritm liniar.*

Input. Numărul de vârfuri n ale poligonului, vârfurile poligonului: $P_1 = (x_{P_1}, y_{P_1}), P_2 = (x_{P_2}, y_{P_2}), \ldots, P_n = (x_{P_n}, y_{P_n})$ (în această ordine), punctul Q din $\mathbb{R}^2$.

Output. Programul afişează poziţia relativă a punctului Q faţă de poligon (în interior, în exterior, pe laturi).

Exemplu. $n = 12$, $P_1 = (0,6)$, $P_2 = (0,0)$, $P_3 = (6,0)$, $P_4 = (6,6)$, $P_5 = (2,6)$, $P_6 = (2,2)$, $P_7 = (4,2)$, $P_8 = (4,5)$, $P_9 = (5,5)$, $P_{10} = (5,1)$, $P_{11} = (1,1)$, $P_{12} = (1,6)$, $Q_1 = (3,4)$. Pentru Q_1 afişează `punctul este în interiorul poligonului`. Pentru acelaşi poligon şi $Q_2 = (7,3)$ afişează `punctul este în exteriorul poligonului`. Pentru acelaşi poligon şi $Q_3 = (3,2)$ afişează `punctul este pe una dintre laturile poligonului`.

2. (1p) *Monotonia unui poligon în raport cu axele de coordonate.*

Input. Numărul de vârfuri n, vârfurile poligonului: $P_1 = (x_{P_1}, y_{P_1}), P_2 = (x_{P_2}, y_{P_2}), \ldots, P_n = (x_{P_n}, y_{P_n})$ (în această ordine) din $\mathbb{R}^2$.

Output. Programul stabileşte dacă poligonul este x-monoton şi dacă este y-monoton.

Precizare. Pentru testare, $P_1P_2 \ldots P_n$ reprezintă un poligon parcurs în sens trigonometric (acest lucru nu mai trebuie verificat). Algoritmul va avea complexitatea-timp liniară.

Exemple. (i) $n = 6$, $P_1 = (4,5)$, $P_2 = (5,7)$, $P_3 = (5,9)$, $P_4 = (2,5)$, $P_5 = (4,2)$, $P_6 = (6,3)$. Se afişează `Poligonul nu este x-monoton. Poligonul este y-monoton`.

(ii) $n = 8$, $P_1 = (8,7)$, $P_2 = (7,5)$, $P_3 = (4,5)$, $P_4 = (3,9)$, $P_5 = (0,1)$, $P_6 = (5,2)$, $P_7 = (3,3)$, $P_8 = (10,3)$. Se afişează `Poligonul nu este x-monoton. Poligonul nu este y-monoton`.

3. (0,5p) *Poziţia unui punct faţă de cercul circumscris unui triunghi.*

Input. Patru puncte A, B, C, D din $\mathbb{R}^2$ cu A, B, C necoliniare.

Output. Programul afişează poziţia relativă a punctului D faţă de cercul circumscris triunghiului ΔABC (în interior, în exterior, pe cerc).

4. (0,5p) *Muchii ilegale.*

Input. Patru puncte A, B, C, D din $\mathbb{R}^2$, reprezentând vârfurile unui patrulater convex.

Output. Programul indică dacă una dintre muchiile AC sau BD este ilegală.

Precizare. Pentru testare, se prespune că patrulaterul $ABCD$ este convex, sensul de parcurgere fiind cel trigonometric (acest lucru nu mai trebuie verificat).

Algoritmi avansaţi

Laborator 7 (săpt. 13 şi 14)

1. (1p) *Intersecţii de semiplane orizontale şi verticale.*

Input. Numărul n de semiplane, coeficienţii care determină inecuaţia fiecărui semiplan. Astfel, pentru semiplanul i ($i = 1, \ldots, n$) sunt citiţi coeficienţii a_i, b_i, c_i corespunzători unei inecuaţii de forma $a_i x + b_i y + c_i \leq 0$.

Output. Programul afişează natura intersecţiei, conform următoarelor situaţii: (a) intersecţie vidă; (b1) intersecţie nevidă şi nemărginită; (b2) intersecţie nevidă şi mărginită.

Precizare. Pentru testare semiplanele vor fi orizontale şi verticale (ambele situaţii sunt posibile, iar acest lucru nu mai trebuie verificat). Algoritmul va avea complexitatea-timp liniară.

Exemple. (i) $n = 3$, $(a_1, b_1, c_1) = (1, 0, -1)$, $(a_2, b_2, c_2) = (-1, 0, 2)$, $(a_3, b_3, c_3) = (0, 1, 3)$. Cele trei semiplane au inecuaţiile $x - 1 \leq 0$, $-x + 2 \leq 0$, respectiv $y + 3 \leq 0$. Inecuaţiile pot fi rescrise $x \leq 1$, $x \geq 2$, $y \leq -3$. Se afişează `intersecţia este vidă`.

(ii) $n = 3$, $(a_1, b_1, c_1) = (-1, 0, 1)$, $(a_2, b_2, c_2) = (1, 0, -2)$, $(a_3, b_3, c_3) = (0, 1, 3)$. Cele trei semiplane au inecuaţiile $-x + 1 \leq 0$, $x - 2 \leq 0$, respectiv $y + 3 \leq 0$. Inecuaţiile pot fi rescrise $x \geq 1$, $x \leq 2$, $y \leq -3$. Se afişează `intersecţia este nevidă, nemărginită`.

(iii) $n = 4$, $(a_1, b_1, c_1) = (-1, 0, 1)$, $(a_2, b_2, c_2) = (1, 0, -2)$, $(a_3, b_3, c_3) = (0, 1, 3)$, $(a_4, b_4, c_4) = (0, -2, -8)$. Cele patru semiplane au inecuaţiile $-x + 1 \leq 0$, $x - 2 \leq 0$, $y + 3 \leq 0$, respectiv $-2y - 8 \leq 0$. Inecuaţiile pot fi rescrise $x \geq 1$, $x \leq 2$, $y \leq -3$, $y \geq -4$. Se afişează `intersecţia este nevidă, mărginită`.

2. (2p) *Poziţia unui punct faţă de semiplane orizontale şi verticale.*

Input. Coordonatele x_Q, y_Q ale punctului Q, numărul n de semiplane, coeficienţii care determină inecuaţia fiecărui semiplan. Pentru semiplanul i ($i = 1, \ldots, n$) sunt citiţi coeficienţii a_i, b_i, c_i corespunzători unei inecuaţii de forma $a_i x + b_i y + c_i \leq 0$.

Output. Programul stabileşte (a) dacă punctul Q este situat în interiorul unui dreptunghi ale cărui vârfuri sunt exact intersecţii ale dreptelor suport ale semiplanelor iar laturile sunt incluse în dreptele suport corespunzătoare; (b) dacă răspunsul de la (a) este afirmativ, determină valoarea minimă a ariilor tuturor dreptunghiulor cu această proprietate.

Precizare. Pentru testare semiplanele vor fi orizontale şi verticale (ambele situaţii sunt posibile, iar acest lucru nu mai trebuie verificat). Algoritmul va avea o complexitate-timp cât mai bună.

Exemple. (i) $Q = (1.5, -4)$, $n = 3$, $(a_1, b_1, c_1) = (-1, 0, 1)$, $(a_2, b_2, c_2) = (1, 0, -2)$, $(a_3, b_3, c_3) = (0, 1, 3)$. Cele trei semiplane au inecuaţiile $-x + 1 \leq 0$, $x - 2 \leq 0$, respectiv $y + 3 \leq 0$. Inecuaţiile pot fi rescrise $x \geq 1$, $x \leq 2$, $y \leq -3$. Puncte de intersecţie între dreptele suport sunt $(1, -3)$, $(2, -3)$. Nu există un dreptunghi (nedegenerat) ale cărui vârfuri să fie exact intersecţii ale dreptelor suport ale semiplanelor date, chiar dacă punctul Q aparţine intersecţiei semiplanelor. Se afişează `(a) nu există un dreptunghi cu proprietatea cerută`.

(ii) $Q = (0, 0)$, $n = 4$, $(a_1, b_1, c_1) = (-1, 0, 1)$, $(a_2, b_2, c_2) = (1, 0, -2)$, $(a_3, b_3, c_3) = (0, 1, 3)$, $(a_4, b_4, c_4) = (0, -2, -8)$. Cele patru semiplane au inecuaţiile $-x + 1 \leq 0$, $x - 2 \leq 0$, $y + 3 \leq 0$, respectiv $-2y - 8 \leq 0$. Inecuaţiile pot fi rescrise $x \geq 1$, $x \leq 2$, $y \leq -3$, $y \geq -4$. Există un dreptunghi cu vârfurile date de intersecţia dreptelor suport (vârfurile sunt $(1, -3), (1, -4), (2, -4), (2, -3)$, dar punctul Q nu este situat în interiorul acestuia. Se afişează `(a) nu există un dreptunghi cu proprietatea cerută`.

(iii) $Q = (1.25, -3.5)$, $n = 4$, $(a_1, b_1, c_1) = (-1, 0, 1)$, $(a_2, b_2, c_2) = (1, 0, -2)$, $(a_3, b_3, c_3) = (0, 1, 3)$, $(a_4, b_4, c_4) = (0, -2, -8)$. Cele patru semiplane au inecuaţiile $-x + 1 \leq 0$, $x - 2 \leq 0$, $y + 3 \leq 0$, respectiv $-2y - 8 \leq 0$. Inecuaţiile pot fi rescrise $x \geq 1$, $x \leq 2$, $y \leq -3$, $y \geq -4$. Există un singur dreptunghi cu vârfurile date de intersecţia dreptelor suport (vârfurile sunt $(1, -3), (1, -4), (2, -4), (2, -3)$, aria este 1), iar punctul Q este situat în interiorul acestuia. Se afişează `(a) există un dreptunghi cu proprietatea cerută, (b) valoarea minimă a ariilor dreptunghiurilor cu proprietatea cerută este 1`.

(iv) $Q = (2, 1)$, $n = 11$, $(a_1, b_1, c_1) = (-1, 0, -1)$, $(a_2, b_2, c_2) = (0, -3, -6)$, $(a_3, b_3, c_3) = (0, 2, -6)$, $(a_4, b_4, c_4) = (1, 0, -3)$, $(a_5, b_5, c_5) = (0, 1, -2)$, $(a_6, b_6, c_6) = (2, 0, -10)$, $(a_7, b_7, c_7) = (0, -1, -3)$, $(a_8, b_8, c_8) = (-4, 0, 0)$, $(a_9, b_9, c_9) = (-1, 0, 1)$, $(a_{10}, b_{10}, c_{10}) = (0, -1, -1)$, $(a_{11}, b_{11}, c_{11}) = (1, 0, -4)$. Inecuaţiile semiplanelor sunt, respectiv: $x \geq -1$, $y \geq -2$, $y \leq 3$, $x \leq 3$, $y \leq 2$, $x \leq 5$, $y \geq -3$, $x \geq 0$, $x \geq 1$, $y \geq -1$, $x \leq 4$. Există mai multe dreptunghiuri ca în enunţ astfel ca punctul Q să fie situat în interiorul acestora. Valoarea minimă a ariilor acestora este 8. Se afişează `(a) există un dreptunghi cu proprietatea cerută, (b) valoarea minimă a ariilor dreptunghiurilor cu proprietatea cerută este 8`.