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. (1p) *Roby*

Soluţiile pentru această problemă pot fi testate pe https://cms.fmi.unibuc.ro/

Descriere. Roby este un aspirator-robotel care are sarcina de a face curat intr-o camera. Robotelul pleaca dintr-un punct de start P_1 si apoi urmeaza un traseu care este o linie poligonala $P_1P_2 \dots P_nP_1$, la final robotelul oprindu-se in P_1. Fiecare punct P_i este indicat prin coordonatele sale (x_i, y_i). In fiecare punct P_i robotelul trebuie sa vireze la stanga sau la dreapta sau sa continue sa mearga pe aceeasi dreapta. La final, pe langa curatarea camerei, Roby trebuie sa indice numarul total de viraje la stanga, numarul total de viraje la dreapta si numarul de situatii in care a ramas pe aceeasi dreapta. Ajutati-l pe Roby sa isi finalizeze cu bine sarcina, indicand cele trei numere.

Date de intrare. Datele de intrare se vor citi de la tastatură. Datele conțin pe prima linie un număr natural n. Pe urmatoarele n linii se afla perechi de numere intregi, reprezentand coordonatele punctelor $P_1, P_2, \dots, P_n$, in aceasta ordine. Pentru fiecare i, pentru punctul P_i sunt indicate pe aceeasi linie coordonatele x_i si y_i, separate prin spatiu.

Date de iesire. Se vor afișa pe o singură linie, separate prin spatiu, numarul total de viraje la stanga, numarul total de viraje la dreapta si numarul de situatii in care ramas pe aceeasi dreapta (in aceasta ordine).

Restrictii si precizari.

- $3 \leq n \leq 1\,000$.
- $-10\,000 < x_i, y_i < 10\,000$, $\forall i = \overline{1, n}$.
- Cazul de coliniaritate include situatiile urmatoare: (i) robotelul continua deplasarea in acelasi sens, (ii) robotelul schimba sensul deplasarii ramanand pe aceeasi dreapta, (iii) cel putin doua dintre punctele pentru care se realizeaza testarea coincid.

Exemplu.
Input

```
7
1 1
2 2
2 0
3 0
4 0
5 0
6 0
```

Output

```
2 1 3
```

Explicatie Traseul parcurs de Roby are in total `6` viraje: `2` la stanga (in punctele P_3 si P_7), `1` la dreapta (in P_2) si are `3` puncte in care continua drept inainte (in P_4, P_5 si P_6). In P_1 nu este realizat niciun viraj, deoarece robotelul se opreste.

3. (0,5p) *Algoritm cu complexitate-timp liniară pentru frontiera acoperirii convexe a unui poligon stelat 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 (aceste ipoteze nu mai trebuie verificate). Un poligon este stelat dacă există un punct M în interiorul său astfel ca, oricum se alege un punct X pe laturile poligonului sau vârf al acestuia, segmentul $[MX]$ este conţinut în întregime în interiorul poligonului. Algoritmul va avea complexitatea-timp liniară.

4. (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.

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