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.*