Examen Algoritmi Avansaţi. 14.06.2023.

1. Fie $A = (1, 1)$, $B = (4, 7)$, $M_\alpha = (-1 + \alpha, -3 + 2\alpha)$ ($\alpha \in \mathbb{R}$). Arătaţi că punctele A, B şi M_α sunt coliniare pentru orice α şi determinaţi valorile lui α pentru care $r(A, M_\alpha, B) > 0$. **(10p)**

2. Fie punctele $P_1 = (-5, 3)$, $P_2 = (-4, 3)$, $P_3 = (-3, 3)$, $P_4 = (-2, 4)$, $P_5 = (-1, 6)$, $P_6 = (1, 2)$, $P_7 = (2, 1)$, $P_8 = (3, 2)$. (i) Alegeţi un punct P_9 pe dreapta $x = 5$, astfel ca P_8 să nu aparţină frontierei acoperirii convexe a lui $\mathcal{M}$, unde $\mathcal{M} = \{P_1, P_2, \ldots, P_8, P_9\}$. Justificaţi alegerea făcută! **(4p)** (ii) Detaliaţi cum evoluează lista $\mathcal{L}_i$ a vârfurilor care determină marginea inferioară a frontierei acoperirii convexe a lui $\mathcal{M}$, obţinută pe parcursul Graham's scan, varianta Andrew (punctul P_9 este cel ales la (i)). Justificaţi! **(6p)**

3. Daţi exemplu de mulţimi $\mathcal{M}_1, \mathcal{M}_2 \subset \mathbb{R}^2$, fiecare cu 5 puncte, astfel ca, pentru fiecare dintre ele, diagrama Voronoi asociată să conţină exact 5 semidrepte, iar diagrama Voronoi asociată lui $\mathcal{M}_1 \cup \mathcal{M}_2$ să conţină exact 3 semidrepte. Justificaţi! **(5p)** Justificaţi câte muchii şi câte triunghiuri are o triangulare a mulţimii $\mathcal{M}_1 \cup \mathcal{M}_2$. **(5p)**

4. Fie semiplanele $H_1 : -y - 1 \leq 0, H_2 : -x - 2 \leq 0, H_3 : -x + y - 3 \leq 0$. (i) Alegeţi două semiplane H_4 şi H_5 astfel încât să fie îndeplinite simultan condiţiile: (i) intersecţia $D = H_1 \cap H_2 \cap H_3 \cap H_4 \cap H_5$ este un trapez dreptunghic şi punctul $(-2, 1)$ este vârf al acestuia **(5p)**; (ii) considerând problema de programare liniară dată de semiplanele H_1, H_2, H_3, H_4, H_5 şi de funcţia obiectiv $f(x, y) = -y$, această problemă admite un unic punct optim (punct de maxim) **(5p)**. Justificaţi şi desenaţi!

5. Fie punctele $A = (2, 3), B = (0, 1), C = (1, 0), D = (-1, -1), E = (2, -4), F = (5, \alpha)$, unde $\alpha \in \mathbb{R} \setminus \{-7, 6\}$. Fie $\mathcal{P} = ABCDEF$. Discutaţi, în funcţie de α: (i) numărul de vârfuri convexe/concave ale poligonului $\mathcal{P}$ **(3p)**; (ii) numărul de vârfuri principale ale lui $\mathcal{P}$ **(5p)**. Explicaţi de ce au fost excluse valorile -7 şi 6 **(2p)**.

6. În planul cartezian Oxy se consideră n triunghiuri $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$. Fiecare triunghi este isoscel şi are o latură orizontală de lungime 8. Pentru fiecare triunghi sunt indicate coordonatele vârfurilor. Se presupune că, pentru toate triunghiurile, acestea sunt întregi şi nu există două vârfuri cu aceeaşi abscisă. Notăm cu σ_{ij} aria intersecţiei triunghiurilor $\mathcal{T}_i$ şi $\mathcal{T}_j$. Descrieţi succint (nu este necesar să detaliaţi calculele matematice, este suficient să le explicaţi pe scurt) un algoritm cât mai eficient care să determine $\max\limits_{i,j=1,\ldots,n}(\sigma_{ij})$. Justificaţi şi exemplificaţi! **(10p)**

7. Daţi exemplu de o problemă de decizie care să fie în clasa **NP** dar nu şi în clasa **NPC**. Presupunem că $P \neq NP$. Justificați-vă pe scurt alegerea făcută. **(5p)**

8. Se dau n obiecte, fiecare dintre ele fiind caracterizat de o valoare, respectiv de o probabilitate (valoare subunitară) de a putea fi transportate intact. Se doreşte alegerea unor obiecte dintre cele n pentru efectuarea unui transport de o valoare cât mai mare, dar cu o probabilitate de cel puțin P ca întreg conținutul să ajungă intact la destinație.

Cerințe: În elaborarea unui algoritm genetic pentru rezolvarea acestei probleme

a) Descrieți ce codificare ati folosi pentru a obtine un cromozom. (Care este lungimea cromozomului? Ce ar reprezenta valoarea fiecării gene?) **(5p)**

b) Descrieți cum ați modela o funcție de fitness pentru această problemă **(5p)**

c) Pentru $n = 8$, $P = 1/8$ şi obiectele cu valoarea, respectiv probabilitatea de a ajunge întregi la destinație sunt $(4, 4/5), (6, 3/5), (3, 4/5), (7, 1/3), (2, 7/10), (10, 1/3), (5, 3/4), (3, 2/3)$. Generati aleator 2 cromozomi (mentionand valoarea fiecaruia dintre ei) si ilustrati pe această pereche operația de crossing-over (recombinare). **(5p)**

9. Fie algoritmul de rezolvare a Problemei de **"Load Balance"** prezentat în cadrul **Cursului 2**, *slide-ul 19*. Acest algorim procesează fiecare task imediat după ce acesta sosește, fără o preprocesare a acestora. În cadrul cursului am arătat că acest algoritm, in forma generală a problemei, este ***2-aproximativ***. Considerăm acuma că rulăm acest algoritm pe o formă restricționată a problemei:

Avem 5 mașini de lucru care trebuie să proceseze n task-uri. Fiecare task i durează t_i unități de timp. În plus, timpul necesar pentru întregul proiect (suma timpurilor tuturor task-urilor) este $T = \sum_{i=1}^{n} t_i = 400$ iar fiecare task i are timpul de lucru limitat la $1 \leq t_i \leq 60$.

Cerință:

a) Arătați că pe un astfel de set de date algoritmul descris în curs este **1.75-aproximativ**. **(5p)**

b) Arătați că pe un astfel de set de date algoritmul descris în curs **nu** este **1.55-aproximativ**. **(5p)**