Tutoriat-1- Programarea Algoritmilor

Exercitii pentru intelegerea limbajului Python si eventual introducere in Metoda Greedy.

Cursul cel mai probabil va urma metodele de progamare Greedy, Divide et Impera, Programare Dinamica, Algoritmi genetici – in ordinea asta dupa cursurile introductive Python.

(I) Se citesc de la tastatura n (natural pozitiv, >3) si n numere intregi care se vor stoca intr-o lista. Afisati cele mai mici 3 numere citite.

(II) Scrieti o functie care are ca input doua cuvinte si returneaza True, daca ele sunt anagrame sau False altfel. (Doua cuvinte sunt anagrame daca se pot scrie cu aceleasi litere prin schimbarea ordinii)

(III) Avand urmatoarea lista de cuvinte: `["haha", "poc", "Poc","haha", "POC", "haHA", "hei", "hey", "HahA", "poc", "Hei"]` , sa se foloseasca un dictionar pentru a numar de cate ori apare fiecare cuvant in lista (indiferent daca este scris cu litere mici sau mici si mari)

(IV) Scrieti o functie care primeste ca parametru n(natural pozitiv) si afiseaza o matrice n * n ale carei elemente sunt egale cu I+J unde I

si J sunt indicii elementului(linie/coloana). Exemplu: n = 2 va genera (0,1) (1,2) in ordine (I=0,j=0) (I=0,j=1) (I=1,j=0) (I=1,j=1) (Folositi list comprehension daca stiti) .

(V) Pentru matricea generata la IV, scrieti o functie care primeste ca parametrii acea matrice si un numar n si care afiseaza doar liniile a caror suma este mai mare de n. (Tot cu list comprehension)

(VI) Implementati metoda Bubble Sort pentru o lista de numere intregi.

(VII) Implementati cautarea binara. Ce complexitate are?

(VIII) Avand un interval si o lista de intervale, determinati care intervale din lista se pot folosti pentru “a acoperi” primul interval astfel incat numarul de intervale ales sa fie minim. Daca nu se poate, afisati un mesaj corespunzator. Exemplu:

[1, 10] si [[-3,4], [2,6], [5,12], [3,6] , [4,8]] -> [-3,4] , [4,8], [5,12]

[2,5] si [[-1,3], [-2,2], [5,8]] -> nu se poate