Programare Algoritmică - Examen 20.01.2022

Muscalu Diana - Grupa 144

SUBIECTUL I

a)
```
def pareimpare(*ls):
    n = len(ls)
    d = { }
    for i in range(n):
        pare = [x for x in ls[i] if x % 2 == 0]
        impare = [x for x in ls[i] if x % 2 == 1]
        l = [ ]
        l.append(impare)
        l.append(pare)
        d[i] = l
    return d
```

b)
```
numere = [x for x in range(100, 1000) if x % 5 != 0
          and x % 10 == x // 100]
```

Muscalu Diana - Grupa 144

SUBIECTUL I

c) $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + 1$

$= 2\left[2\,T\left(\frac{n}{2^2}\right) + 1\right] + 1$

$= 2^2 \cdot T\left(\frac{n}{2^2}\right) + 2 + 1$

$= 2^3 \cdot T\left(\frac{n}{2^3}\right) + 2^2 + 2 + 1$

.....

$= 2^k \cdot T\left(\frac{n}{2^k}\right) + 2^{k-1} + 2^{k-2} + \dots + 1$

$n = 2^k \Rightarrow k = \log_2 n$

$T(n) = n \cdot T\left(\frac{n}{2^{\log_2 n}}\right) + 2^{k-1} + \dots + 2 + 1$

$T(n) = n \cdot 1 + 2^{k-1} + \dots + 2 + 1$

$\Rightarrow T(n) = O(n)$

Muscalu Diana - Grupa 144

SUBIECTUL II

```
n = int(input())
h = int(input())
cs = [ ]
for i in range(n):
    s = input().split()
    nume = s[0] + " " + s[1]
    inaltime = int(s[2])
    c = [nume, inaltime]
    cs.append(c)
cs.sort(key = lambda t: (t[1]))
rez = [ ]
for i in range(n-1):
    if cs[i+1][1] - cs[i][1] < h:
        if cs[i][0] not in rez and cs[i+1][0] not in rez:
            rez.append(cs[i][0])
            rez.append(cs[i+1][0])
print(len(rez)//2)
for i in range(0, len(rez), 2):
    print(rez[i], rez[i+1], sep=", ")
```

Muscalu Diana - Grupa 144

SUBIECTUL II

Metoda Greedy construiește o soluție optimă alegând la fiecare pas cel mai bun element.

Astfel pt. a rezolva problema am sortat elementele după înălțime în ordine crescătoare. Deci alăturat vom avea două persoane care au diferența cea mai mică de înălțime. Parcurgem pe rând câte două persoane de-o dată le verificăm diferența, iar dacă condiția este îndeplinită îl adăugăm în soluție

Corectitudine:

Considerăm mulțimea ordonată crescător după înălțimile elevilor.

$M = \{i_1, i_2 \dots i_n\}$ deci avem $h_{i_1} \leq h_{i_2} \leq \dots \leq h_{i_n}$ (1)

Astfel primul elev din M va avea valoarea h_{i_1} minimă dintre toate valorile h_{ik} unde $k = \overline{1,n}$

~~Dem. că mereu va ∃ o sol optimă care îl conține pe i_1.~~

Dem. că mereu va ∃ o sol. optimă care îl conține pe primul element i_k cu $k = \overline{1,n}$ care e compatibil cu următorul element i_{k+1}. îl vom nota i_s

Fie $A \subset M$ o sol. optimă a pb. unde $A = \{j_1, \dots j_k\}$

Pp. că această sol. nu îl conține pe i_s. $\Rightarrow h_{j_1} \leq h_{j_2}$ (2)

Fie $A' = \{i_s, j_1 \dots j_k\}$

Din (1) (2) $\Rightarrow h_{is} \leq h_{j_1}$ deci primul elev soluție i_s va fi compatibil cu $j_2, j_3 \dots j_k$

Muscalu Diana - Grupa 144

Deci toți elevii din A' sunt compatibili $|A'|=|A| \Rightarrow A'$ sol. optimă

Am. dem. că sol. opt. conține i_s. În cont. alegem elevii compatibili cu i_s

Dacă A e sol. opt. pt. M, vom dem că $A'' = A - \{i_s\}$ Constr. cu algoritmul nostru e o sol. opt. pt. $M' = M - \{j \mid h_j \leq h_{i_s}\}$

Pp. că A'' nu e sol. opt. $\Rightarrow \exists B$. o. î. $|B|>|A''|$, dar atunci $|B \cup \{i_s\}| > |A|$ ↯ $\Rightarrow$ A sol. optimă

Complexitate:

Problema presupune citire + sortare + parcurgere pt. construirea mulțimii $\Rightarrow O(n \log n)$

SUBIECTUL IV

a) Tehnica Backtracking generează toate soluțiile și le formează element cu element, încercând să vadă dacă elem. curent poate fi pus în soluție

Reprezentarea soluției: $(x_1 x_2 \dots x_k)$ unde x_k aparține fie lui Z fie lui S

Condiții finale: k să fie egal cu n

Condiții de continuitate x_k: elementul curent pe care vrem să îl adăugăm (x_k) să nu se regăsească deja în soluție ($x[1:k-1]$)

Muscalu Diana - Grupa 144

SUBIECTUL IV

a)
```
def back(k):
    if k==n:
(*) print(*x, sep="")
    else:
        if T[k]=='e':
            for i in range(len(L)):
                if L[i] not in x:
                    x[k]=L[i]
                    back(k+1)
                    x[k]=0
        else:
            for i in range(len(S)):
                if S[i] not in x:
                    x[k]=S[i]
                    back(k+1)
                    x[k]=0

n=int(input())
T=input()
L=[i for i in input().split()]
S=[i for i in input().split()]
x=[0 for i in range(n)]
back(0)
```

Muscalu Diana - Grupa 144

SUBIECTUL IV

b) (*) if x[0] in "aeiouAEIOU":
        print(*x, sep="")