

Subiectul I

a) def cheie(x):  
 return -x

def medie(numar):

D=0

nr=0

while numar:

D += numar % 10

nr += 1

numar //= 10

return D/nr

def numere(\*numere):

dict = {}

for numar in numere:

if medie(numar) not in dict:

dict[medie(numar)] = [numar]

else:

dict[medie(numar)].append(numar)

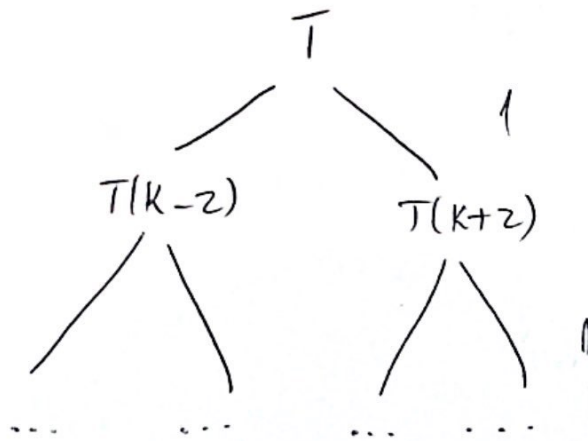
for c in dict.keys():

dict[c] = sorted(dict[c], key=cheie)

return dict

b)  $L = [\text{int}(x) \text{ for } x \text{ in range}(100, 1000)]$

c)



Recursivitatea se va strânge

$$2T(n/2) + 1$$

cu ajutorul teoremei de master

$$\Rightarrow a = 2$$

$$b = 2$$

$$K = 0$$

$$P = 0$$

$$T(n) = aT(n/b) + f(n)$$

Ne aflăm în cazul I, când  $\log_a b > K$  ( $\log_2 2 > 0$ ) A),

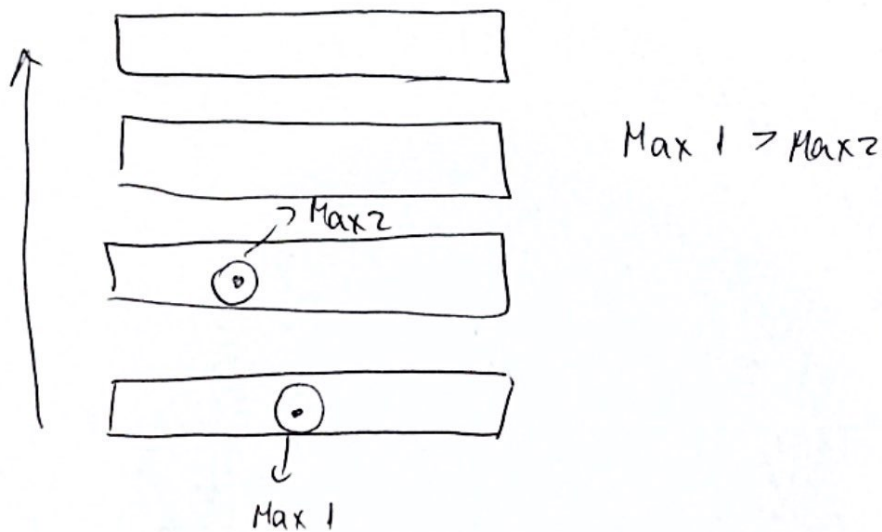
$\Rightarrow$  complexitatea recursivității este

$$O(n^{\log_a b}) \Rightarrow O(n^1) = O(n)$$

## Subiectul 2 - Greedy

Trebuie să găsim cele mai mari valori dintre-un vector  $m \times n$ .

Începem de pe ultimul rând, căutăm valoarea maximă din vectorul respectiv



Parcurgerea pe linii se realizează de jos în sus, luând mereu maximumul de pe linia curentă cu proprietatea că este mai mic decât maximumul de pe linia de jos.

## Subiectul 4 - Backtracking

```
a) def back (poz, s):  
    global ok  
    if s == G:  
        print (*x[1:poz])  
        ok = 1  
    else:  
        for i in range (1, K+1):  
            x[poz] = i  
            print (x)  
            s += x[poz]  
            if abs(x[poz-1] - x[poz]) <= K:  
                if poz != m:  
                    back (poz+1, s)  
            s -= x[poz]
```

```
G, m, K = (int(i) for i in input().split())
```

```
x = [0 for i in range (m+1)]
```

```
ok = 0
```

```
back(1, 0)
```

```
if (ok
```

```
if ok == 0:
```

```
    print("Imposibil")
```

Algoritmul se încadrează în metoda Backtracking deoarece generează toate soluțiile posibile.

Variabilele  $G, m, k$  reprezintă datele problemei:

Variabila „ok” verifică dacă s-a afișat cel puțin o soluție, în caz contrar afișează „Imposibil”.

Variabila  $s$  este o sumă, ce verifică dacă suma componentelor din  $x$  a ajuns la valoarea „ $G$ ”, adică suma totală contă.

„ $x$ ” reprezintă vectorul de soluție