

01.02.2021

Pop Mihai-Daniel

Subiectul 12

Examen scris

Grupa 215

Fundamentele programării

① Sortarea listă de numere : mergesort

```
def merge (aux1, aux2):
```

```
    i = 0
```

```
    j = 0
```

```
    aux3 = []
```

```
    while i < len(aux1) and j < len(aux2):
```

```
        if aux1[i] < aux2[j]:
```

```
            aux3.append(aux1[i])
```

```
            i += 1
```

```
        else:
```

```
            aux3.append(aux2[j])
```

```
            j += 1
```

```
    while i < len(aux1):
```

```
        aux3.append(aux1[i])
```

```
        i += 1
```

```
    while j < len(aux2):
```

```
        aux3.append(aux2[j])
```

```
        j += 1
```

```
    return aux3
```

```
def mergeSort (lista):
```

```
    if len(lista) == 1:
```

```
        return lista
```

```
    mij = len(lista) // 2
```

```
    aux1 = mergeSort(lista[:mij])
```

```
    aux2 = mergeSort(lista[mij:])
```

```
    return merge(aux1, aux2)
```


② Specificații:

" " "

Funcția dată creează o listă descrescătoare de ~~la~~
~~un număr dat~~ cu un număr dat de elemente, puse
în listă în ordine descrescătoare, consecutive, strict mai mici
decît numărul dat; ea calculează suma elementelor
 n - număr întreg (funcția calculează suma primilor $n-1$ ^{numere})
reținându-și suma elementelor listei create, această sumă
fiind calculată succesiv și reținută în ultimul element al
listei.

Ridică excepție de tip `ValueError` dacă numărul n
dat este mai mic sau egal cu 0.

" " "

Teste:

def tests():

assert (f2(5) == 10)

assert (f2(1) == 0)

assert (f2(2) == 1)

try:

f2(0)

assert False

except ValueError:

assert True

try: ~~f2~~

f2(-5)

assert False

except ValueError:

assert True

③ Complexitate

time: Caz favorabil: - lista contine un singur element $\Rightarrow T(n) = 1$

Caz defavorabil: - lista nu contine elemente egale cu 0 $\Rightarrow T(n) = n$

Caz mediu: - lista poate contine pe orice pozitie un element egal cu 0.

n - lungimea listei

$$T(n) = \sum_{i=1}^n i = 1+2+3+\dots+n = \frac{n(n+1)}{2} \in \Theta(n^2) \Rightarrow$$

\Rightarrow Overall complexity: $T(n) \in O(n^2)$

spatiu: complexitate $T(n)$ - functie ^{mat.} in-place
la fiecare pas se modifica lista data cu un elem.

④ Divide et impera - nr. de nr. negative

def nr_neg(l, st, dr):

if dr == st:

return 0

if dr - st == 1:

if l[st] < 0:

return 1

else:

return 0

mij = (st + dr) // 2

return nr_neg(l, st, mij) + nr_neg(l, mij, dr)

x = nr_neg(l, 0, len(l))

⑤ Soluție backtracking - fără implementare

Toate sublistele care conțin doar nr pare sau doar impare

Ne dorim subliste :

Soluție candidat :

$x_i \in \{0, 1, \dots, n-1\}$, n - lungimea listei

$X = (x_0, x_1, \dots, x_k)$, ~~$x_i \in \text{lista data}$~~

Condiție consistent :

l - lista dată

$X = (x_0, x_1, \dots, x_k)$ - l consistent dacă ~~$x_i \% 2 \neq x_j \% 2$~~
pt $\forall i \neq j$ și $x_i \neq x_j$ $l[x_i] \% 2 = l[x_j] \% 2$

Condiție soluție

$X = (x_0, x_1, \dots, x_k)$ - l soluție dacă l consistent și
 $k \leq n-1$, n - lungimea listei date