

# Pismeni zadatak - pretraga sa vraćanjem i dinamičko programiranje

*Napomena:* Prvi zadatak se radi na papiru, ostali se programiraju u C#.

**Zadatak 1** (10 poena) *Problem ranca kapaciteta 16 bez ograničenja.*

p:	p1	p2	p3	p4	p5	p6	p7	p8
v:	1	2	8	14	6	13	7	10
x:	1	2.2	12	15	7	15	8	13.5

Veza između rešenja problema manjih i većih dimenzija je sledeća (sa  $a[c]$  je obeleženo optimalno rešenje za kapacitet  $c$ ):

$$a[c] = \begin{cases} 0, & c = 0 \\ \max\{a[c - v[i]] + x[i]\}, i = 1, \dots, N, v[i] \leq c, & c > 0 \end{cases} \quad (1)$$

**Zadatak 2** (15 poena) *Implementirati program koji generše sve dekadne brojeve sa  $N$  cifara (bez cifre 0), takve da je broj zbir cifara veći od  $D$ , a manji od  $G$ . Korisnik unosi  $N$ ,  $D$  i  $G$ . Nije potrebno čuvati dekadne brojeve, dovoljno je samo ispisati ih. Test primer (redosled ispisa ne mora biti isti kao u test primeru):*

```
3 10 20
119
129
139
...
```

**Zadatak 3** *Edit rastojanje za data dve reči  $x$  i  $y$ , predstavlja najmanji broj operacija izmene, brisanja ili dodavanja slova, koji od prve reči drugu. Npr., za reči  $x$ ="tekst" i  $y$ ="test" edit rastojanje je 1, zato što je potrebno samo ukloniti slovo  $k$  iz prve reči. Za reči  $x$ ="formirati" i  $y$ ="normiran", edit rastojanje se zasniva na zamenama  $f \rightarrow n$ ,  $t \rightarrow n$ , i brisanja slova  $i$  iz prve reči, tako da je edit rastojanje 3. Napisati program koji: a) nalazi edit rastojanje dinamičkim programiranjem (13 poena) i b) rekurzivno (12 poena).*

Ako sa  $a[i, j]$  označimo edit rastojanje za prvih  $i$  slova prve reči i prvih  $j$  slova druge reči, onda je relacija između rešenja problema i potproblema data na sledeći način (ova relacija se može iskoristiti i kod rekurzivnog rešenja):

$$a[i, j] = \begin{cases} j, & i = 0 \\ i, & j = 0 \\ \min\{a[i, j - 1] + 1, a[i - 1, j] + 1, a[i - 1, j - 1] + \mathbb{1}_{x[i]=y[j]}\}, & i > 0, j > 0 \end{cases} \quad (2)$$

*Test primer:*

		P	O	L	Y	N	O	M	I	A	L
E	0	1	2	3	4	5	6	7	8	9	10
X	1	1	2	3	4	5	6	7	8	9	10
P	2	2	2	3	4	5	6	7	8	9	10
O	3	2	3	3	4	5	6	7	8	9	10
N	4	3	2	3	4	5	5	6	7	8	9
E	5	4	3	3	4	4	5	6	7	8	9
N	6	5	4	4	4	5	5	6	7	8	9
T	7	6	5	5	5	4	5	6	7	8	9
I	8	7	6	6	6	5	5	6	7	8	9
A	9	8	7	7	7	6	6	6	6	7	8
L	10	9	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	7	6