Programiranje I — 4. domača naloga

Rok za oddajo: nedelja, 4. december 2016, ob 23:55

Modulska aritmetika

Uvod

Za podano praštevilo p, imenovano *modul*, definirajmo množico $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$. Za števila iz te množice sta operaciji *modulskega seštevanja* ($\oplus$) in *modulskega množenja* ($\otimes$) definirani takole:

$$a \oplus b = (a + b) \bmod p$$
$$a \otimes b = ab \bmod p$$

Zapis $s \bmod t$ predstavlja ostanek pri deljenju števila s s številom t.

Modulska nasprotna vrednost števila a je število $\overline{a}$, za katero velja $a \oplus \overline{a} = 0$. *Modulska obratna vrednost* števila $a \neq 0$ je število $a^\star$, za katero velja $a \otimes a^\star = 1$. Sedaj lahko definiramo tudi operaciji *modulskega odštevanja* ($\ominus$) in *modulskega deljenja* ($\oslash$):

$$a \ominus b = a \oplus \overline{b}$$
$$a \oslash b = a \otimes b^\star \text{ (pri pogoju } b \neq 0)$$

Na primer, elementi množice $\mathbb{Z}_7$ se med sabo modulsko seštevajo, odštevajo, množijo in delijo takole:

$\oplus$	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

$\ominus$	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	0	6	5	4	3	2	1
1	1	0	6	5	4	3	2
2	2	1	0	6	5	4	3
3	3	2	1	0	6	5	4
4	4	3	2	1	0	6	5
5	5	4	3	2	1	0	6
6	6	5	4	3	2	1	0

$\otimes$	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

$\oslash$	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	-	0	0	0	0	0	0
1	-	1	4	5	2	3	6
2	-	2	1	3	4	6	5
3	-	3	5	1	6	2	4
4	-	4	2	6	1	5	3
5	-	5	6	4	3	1	2
6	-	6	3	2	5	4	1

Modulska potenca $\mathrm{mpow}(a,\, t)$, pri čemer je $a \in \mathbb{Z}_p$, t pa je lahko *poljubno* celo število, je definirana takole:

$$\mathrm{mpow}(a,\, t) = \begin{cases} 1 & \text{pri } t = 0 \\ a \otimes \mathrm{mpow}(a,\, t-1) & \text{pri } t > 0 \\ \mathrm{mpow}(a,\, -t)^\star & \text{pri } t < 0 \end{cases}$$

Na primer, pri $p = 7$ velja $\mathrm{mpow}(5, 0) = 1$, $\mathrm{mpow}(5, 1) = 5$, $\mathrm{mpow}(5, 2) = 5 \otimes 5 = 4$, $\mathrm{mpow}(5, 3) = 5 \otimes 5 \otimes 5 = 6$, $\mathrm{mpow}(5, -1) = 5^\star = 3$, $\mathrm{mpow}(5, -2) = (5 \otimes 5)^\star = 4^\star = 2$ itd.

Definirajmo še *modulski kvadratni koren* (msqrt):

$$\mathrm{msqrt}(a) = \{b \in \mathbb{Z}_p \mid b \otimes b = a\}$$

Za razliko od običajnega kvadratnega korena modulski ni enolično določen, zato ga definiramo kot množico vseh števil, ki pri modulskem množenju s samim seboj dajo podano število. Na primer, pri $p = 7$ velja $\mathrm{msqrt}(0) = \{0\}$, $\mathrm{msqrt}(1) = \{1, 6\}$, $\mathrm{msqrt}(2) = \{3, 4\}$, $\mathrm{msqrt}(3) = \varnothing$, $\mathrm{msqrt}(4) = \{2, 5\}$, $\mathrm{msqrt}(5) = \varnothing$ in $\mathrm{msqrt}(6) = \varnothing$.

Število $n \in \mathbb{Z}_p$ je *multiplikativni generator* množice $\mathbb{Z}_p$, če je množica

$$P(n) = \{\mathrm{mpow}(n,\, i) \mid i \in \{0,\, 1,\, \ldots,\, p-1\}\}$$

enaka množici $\mathbb{Z}_p \backslash \{0\} = \{1,\, 2,\, \ldots,\, p-1\}$. Na primer, pri $p = 7$ velja $P(0) = \{0\}$, $P(1) = \{1\}$, $P(2) = \{1, 2, 4\}$, $P(3) = \{1, 2, 3, 4, 5, 6\}$, $P(4) = \{1, 2, 4\}$, $P(5) = \{1, 2, 3, 4, 5, 6\}$ in $P(6) = \{1, 6\}$. Multiplikativna generatorja množice $\mathbb{Z}_7$ sta torej števili 3 in 5.

Naloga

Napišite razred `Zp`, čigar objekt predstavlja množico $\mathbb{Z}_p$ za podani modul p. Razred naj vsebuje sledeče javno dostopne konstruktorje in metode:[1]

- `public Zp(int modul)` [J1–J10, S1–S50]:

  Izdela objekt, ki predstavlja množico $\mathbb{Z}_{\texttt{modul}}$. V vseh testnih primerih je `modul` praštevilo z intervala [2, 97].

- `public int vrniModul()` [J1, S1–S5]:

  Vrne modul množice `this` (oziroma, če smo natančnejši, modul množice, ki jo predstavlja objekt, na katerega se sklicuje referenca `this`).

- `public String toString()` [J2, S6–S10]:

  Vrne niz oblike `Z_`p, kjer je p modul množice `this`. Na primer, če objekt `this` predstavlja množico $\mathbb{Z}_7$, naj metoda vrne niz `Z_7`.

- `public int vsota(int prvo, int drugo)` [J3, S11–S15]:

  Vrne modulsko vsoto števil `prvo` in `drugo` v okviru množice `this`. V vseh testnih primerih se parametra `prvo` in `drugo` nahajata v intervalu $[0,\, p-1]$, kjer je p modul množice `this`. Ta pripomba velja tudi za ostale metode, razen kjer je izrecno navedeno drugače.

- `public int zmnozek(int prvo, int drugo)` [J4, S16–S20]:

  Vrne modulski zmnozek števil `prvo` in `drugo`.

[1] V oglatih oklepajih so navedeni testni primeri, ki vsebujejo klice pripadajočih konstruktorjev oz. metod.

- `public int nasprotno(int stevilo)` [J5, S21–S25]:

  Vrne modulsko nasprotno vrednost števila `stevilo`.

- `public int razlika(int prvo, int drugo)` [J5, S21–S25]:

  Vrne modulsko razliko števil `prvo` in `drugo`.

- `public int obratno(int stevilo)` [J6, S26–S30]:

  Vrne modulsko obratno vrednost števila `stevilo`. V vseh testnih primerih se parameter `stevilo` nahaja v intervalu $[1, p-1]$.

- `public int kolicnik(int prvo, int drugo)` [J6, S26–S30]:

  Vrne modulski količnik števil `prvo` in `drugo`. V vseh testnih primerih se parameter `prvo` nahaja v intervalu $[0, p-1]$, parameter `drugo` pa v intervalu $[1, p-1]$.

- `public int potenca(int stevilo, long eksponent)` [J7, J10, S31–S35, S48–S50]:

  Vrne modulsko potenco števila `stevilo` na število `eksponent`. V testnih primerih J7 in S31–S35 se parameter `eksponent` nahaja v intervalu $[-10^3, 10^3]$, v primerih J10 in S48–S50 pa v intervalu $[-10^{18}, 10^{18}]$. Parameter `stevilo` se nahaja v intervalu $[1, p-1]$.

- `public int steviloKvadratnihKorenov(int stevilo)` [J8, S36–S40]:

  Vrne moč množice modulskih kvadratnih korenov števila `stevilo`, torej število števil $b \in \mathbb{Z}_p$, za katera je produkt $b \otimes b$ enak številu `stevilo`.

- `public boolean jeMultiplikativniGenerator(int stevilo)` [J9–J10, S41–S47]:

  Vrne `true` natanko v primeru, če je število `stevilo` multiplikativni generator množice `this`. Parameter `stevilo` se nahaja v intervalu $[1, p-1]$.

Namig

Kako bi učinkovito izračunali potenco a^e (oziroma njeno modulsko različico), kjer je eksponent e veliko pozitivno celo število? Če je e sod, se nam a^e splača izračunati kot $(a^{e/2})^2$, saj lahko potenco $a^{e/2}$ izračunamo na enak način kot potenco a^e, le eksponent je za polovico manjši. Lahko podoben trik uporabimo tudi pri lihih eksponentih?

Oddaja naloge

Oddajte datoteko z nazivom `Zp.java`. V prvi vrstici datoteke v komentarju navedite vašo vpisno številko. Če je, denimo, vaša vpisna številka enaka 63160999, mora datoteka izgledati takole:

```
// 63160999

public class Zp {
    ...
}
```

Testiranje

Program `tj.exe` boste tokrat pognali takole:

```
tj.exe <mapa_z_vašim_razredom> <mapa_s_testnimi_razredi> <mapa_z_rezultati>
```

Če si želite postopek testiranja karseda poenostaviti, postavite datoteko `Zp.java` v mapo, kjer se nahajajo testni razredi. Znotraj te mape boste namreč lahko program `tj.exe` pognali preprosto takole:

```
tj.exe
```

To je okrajšava za ukaz

```
tj.exe . . .
```

kar pomeni, da se vse, tudi bodoči rezultati, nahaja v trenutni mapi. Če se vaš program nahaja v isti mapi kot testni razredi, boste testne razrede lahko prevajali in poganjali tudi ročno (npr. `javac Test01.java` in `java Test01` za prvi testni razred).