

Domača naloga 4 - Dekodiranje

1 Opis

Sporočilo sestavljeno iz znakov angleške abecede je lahko zakodirano, tako da vsakemu znaku določimo številko kot:

'A' \rightarrow '1'
'B' \rightarrow '2'
...
'Z' \rightarrow '26'

Dobljen zakodiran niz lahko nato odkodiramo z obratno operacijo. Očitno dobljen niz ni enoličen, saj lahko različne znake združimo na različne načine. Naprimer zakodiran niz "11106" lahko odkodiramo v

- "AAJF", če znake združimo kot (1 1 10 6)
- "KJF", če znake združimo kot (11 10 6).

Poleg tega lahko zakodirano sporočilo vsebuje znak "*", ki lahko predstavlja katerokoli številko od 1 do 9 (vendar ne 0). Vaša naloga bo, da na podlagi zakodiranega niza s vrnete število različnih nizov v katerega ga lahko odkodiramo. Ker je ta številka lahko velika, rezultat vrnite po modulu z $10^9 + 7$.

2 Vhodni podatki

Vhodni podatki vsebujejo le eno vrstico z zakodiranim nizom s dolžine n .

Omejitve vhodnih podatkov:

- $n \leq 10^5$

3 Izhodni podatki

Izpišite le eno vrstico in to je število načinov, da podan zakodirani niz odkodiramo po modulu z $10^9 + 7$.

Časovna zahtevnost: Vaš algoritem mora delovati v $O(n)$.

Prostorska zahtevnost: Vaš algoritem lahko porabi $O(1)$ dodatnega prostora.

4 Primeri

4.1 Primer 1

Vhod	Izhod
*	9

V tem primeru lahko * nadomesti katerokoli izmed 9-ith števk, zato je odgovor 9.

4.2 Primer 2

Vhod	Izhod
1*	18

Sedaj lahko števko 1 odkodiramo v "A", znak * pa v 9 različnih znakov kar nam da 9 načinov. Poleg tega lahko niz 1* odkodiramo v znake, ki so zakodirani v dve števki. Teh je še dodatnih 9 kar da skupaj 18.

4.3 Primer 3

Vhod	Izhod
0*2*3	0

Ker se nobena koda ne začne s števko 0, tega niza ne moremo odkodirati. Zato vrnemo 0.

5 Testni primeri

Na voljo imate več testnih primerov ter pripadajočih izhodov. Vsak testni primer ima vhodne podatke v datoteki `testXY.in` ter izhodne podatke v `testXY.out`. Testni primeri se nahajajo v mapi `Tests`.

S pomočjo priložene skripte `test_runner.py` lahko zaženete enega, ali pa vse testne primere. Datoteko `resitev.py`, kjer ste rešili nalogo prestavite v isto mapo, kjer se nahaja `test_runner.py` ter mapa `Tests`. Skripta `test_runner.py` ima 3 možne argumente:

- `-script` (obvezen): ime oziroma pot do vaše datoteke z rešitvijo
- `-test_nb` (neobvezen): številka testa, ki naj se izvede. Če ni podano se izvedejo vsi testi.
- `-pypath` (neobvezen): Pot do Python interpreterja oz. okrajšava. Če ni podano, se vzame `python3`.

Opozorilo: Na različnih sistemih ima Python lahko različne okrajšave. Na linux in MacOS operacijskih sistemih Python ponavadi zaganjamo z `python3`, medtem ko na Windows operacijskih sistemih ponavadi z `python` (v kolikor je python dodan v okoljsko spremenljivko). V kolikor boste imeli kakšne težave me kontaktirajte.

Odprite terminal v mapi, kjer imate zgornje datoteke. Nato izvedite ukaz:

```
> python3 test_runner.py -script resitev.py -test_nb 0
```

Ob uspešno izvedenem testu se izpiše:

```
> Test test00.in uspešen. Čas: 0.024 sekunde.
```