



## Opisi algoritama

Zadatke, testne primjere i rješenja pripremili: Nikola Dmitrović, Marin Kišić, Josip Klepec, Ivan Lazarić, Ivan Paljak, Stjepan Požgaj, Daniel Paleka i Paula Vidas. Primjeri implementiranih rješenja su dani u priloženim izvornim kodovima.

### Zadatak: Dinamo

Predložio: Nikola Dmitrović

Potrebno znanje: naredba učitavanja i ispisivanja, jednostavna naredba odlučivanja

Na osnovi ulaznih podataka o klubovima s kojima je Dinamo igrao u prva tri kola, naredbom odlučivanja lako možemo provjeriti s kojim je klubovima igrao u četvrtom, petom i šestom kolu.

S kojim je klubovima igrao u tim kolima pronaći ćemo u tekstu zadatka.

*Programski kod (pisan u Python 3):*

```
A = int(input())
C = int(input())
S = int(input())
X = int(input())

if X == 4: print(S)
if X == 5: print(A)
if X == 6: print(C)
```

### Zadatak: Lijepi

Predložio: Nikola Dmitrović

Potrebno znanje: učitavanje unaprijed poznatog broja brojeva zapisanih u istom retku, naredba ponavljanja, određivanje broja znamenki u broju, string

Pretpostavimo da ne znamo opće rješenje ovog zadatka. Pogledajmo u sekciji *Bodovanje* postoji li neka parcijala, neki dio zadatka koji bismo mogli riješiti.

U prvoj parcijali, vrijednoj 18 bodova, dobit ćemo samo jedan par dvoznamenkastih brojeva  $X$  i  $Y$ . Da bi od njih dobili novi broj, očito je da  $X$  trebamo pomnožiti sa 100 i tako dobivenoj vrijednosti pribrojiti  $Y$ . Uočite da nam za ovo treba samo naredba učitavanja dva broja koji su zapisani u istom retku i naredba odlučivanja. Nužno je učitati i broj  $N$ , iako je njegova vrijednost uvijek jedan i nigdje u rješenju zadatka ga ne koristimo. *Programski kod (pisan u Python 3):*

```
# prva parcijala
N = int(input())
X, Y = map(int, input().split()) # Python učitavanje dva broja u istom retku
zbroj = X * 100 + Y
print(zbroj)
```

Ako se odlučimo rješavati drugu parcijalu, vrijednu 22 boda, morat ćemo poznavati i naredbu ponavljanja.  $N$  puta ćemo dobiti po dva dvoznamenkasta broja  $X$  i  $Y$  te ćemo na svakom tom paru ponoviti prethodno opisani algoritam. Naravno, kako imamo  $N$  takvih parova bit će nužno kontinuirano zbrajati nove vrijednosti u posebnoj varijabli `zbroj`.



*Programski kod (pisan u Python 3):*

```
# druga parcijala
N = int(input())
zbroj = 0
for i in range(N):
    X, Y = map(int, input().split())
    novi = X * 100 + Y
    zbroj = zbroj + novi # može i ovako: zbroj += novi
print(zbroj)
```

Da bismo riješili zadatak za sve bodove, moramo za svaki par brojeva  $X$  i  $Y$  kreirati novi broj na način da prvo odredimo koliko znamenki ima broj  $Y$ , broj  $X$  pomnožimo s potencijom broja 10 na broj tih znamenki i tako dobivenoj vrijednosti pribrojimo vrijednost  $Y$ . Broj znamenki u nekom broju možemo odrediti uzastopnim cjelobrovnim dijeljenjem broja s 10 sve dok taj broj ne postane nula. Traženje broja znamenki moramo raditi na kopiji broja  $Y$  jer ćemo u protivnom izgubiti broj  $Y$  koji nam je nužan za nastavak algoritma.

*Programski kod (pisan u Python 3):*

```
# opće rješenje
N = int(input())
zbroj = 0
for i in range(N):
    X, Y = map(int, input().split())
    brznY = 0
    kopija = Y
    while kopija > 0:
        brznY += 1
        kopija //= 10
    zbroj += X * pow(10, brznY) + Y
print(zbroj)
```

Za kraj, riješimo zadatak koristeći stringove. Brojeve  $X$  i  $Y$  ćemo učitati kao “napisane brojeve”, nalijepiti jedan na drugi koristeći operator “konkateniranja” i tako dobiveni string pretvoriti u prirodan broj naredbom `int()`.

*Programski kod (pisan u Python 3):*

```
# string rješenje
N = int(input())
zbroj = 0
for i in range(N):
    X, Y = input().split()
    zbroj += int(X + Y)
print(zbroj)
```



## Zadatak: Trol

Predložio: Ivan Paljak

Potrebno znanje: dokazivanje slutnji, matematika, suma intervala u periodičkom nizu

Ovaj zadatak se na prvi pogled čini pomalo zastrašujuć jer je potrebno riješiti više upita, brojevi u upitima su vrlo veliki, promjene nad elementima niza su relativno kompleksne, itd. No, budući da se radi o trećem zadatku naziva kao što je “trol”, može se naslutiti da se iza svega krije nešto elegantno.

No, krenimo redom. U test podacima vrijednima 10 bodova radimo nad elementima niza koje Marin uopće nije mijenjao (jer su već bili jednoznačenosti). Za osvajanje ovih bodova bilo je dovoljno za svaki upit ispisati vrijednost  $l + (l + 1) + \dots + (r - 1) + r$ .

Za dodatnih 20 bodova ograničenja su bila dovoljno mala da je bilo moguće simulirati postupak opisan u tekstu zadatka. Odnosno, krećući se petljom po brojevima od  $l$  prema  $r$ , bilo je potrebno za svaki od njih odsimulirati Marinove zamjene. Problem pronalaska zbroja znamenki nekog broja relativno je klasičan i dobro opisan u raznoj literaturi (**hint**: Kako dobiti zadnju znamenku nekog broja? Kako tu znamenku obrisati iz broja?). Ovo je rješenje implementirano u izvornom kodu `trol_brute.cpp`.

Za osvajanje svih bodova na ovom zadatku najprije je potrebno uočiti jednu pravilnost. Tu pravilnost možete uočiti na dva načina koja ćemo vam dočarati istinitim događajima koji su se odvijali prilikom sastavljanja zadataka za ovo kolo.

### Prvi način

Autor je ovaj zadatak najprije ispričao Marinu. Marin je iskusan natjecatelj i zna da prijedlog za treći zadatak na HONI-ju ne smije biti pretjerano težak pa je odmah naslutio da se iza kompliciranih operacija krije neka pravilnost. Unutar 3 minute Marin je napisao prethodno opisano rješenje (za 20 bodova) i pogledao što se događa s nizom. Brzo je zaključio da nakon zamjena niz  $A$  postaje periodičan. Preciznije, uočio je da vrijedi  $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, \dots\}$ . Marin nije odmah znao zašto vrijedi ova pravilnost, ali je eksperimentalno ~~dokazao~~ pokazao svoju slutnju.

### Drugi način

Druga osoba kojoj je autor ispričao ovaj zadatak je Stjepan. Stjepan uistinu jest završio preddiplomski studij matematike pa se ubrzo sjetio da je u osnovnoj školi naučio kako je “broj djeljiv s 9 ako mu je zbroj znamenaka djeljiv s 9”. Također je zaključio da vrijedi i jača tvrdnja, da je ostatak koji broj daje pri dijeljenju s 9 jednak ostatku koji zbroj njegovih znamenaka daje pri dijeljenju s 9. Budući da se ostatak nekog broja pri dijeljenju s 9 ne mijenja kada ga zamijenimo zbrojem njegovih znamenaka, a svaka znamenka od 1 do 9 ima jedinstven ostatak pri dijeljenju s 9, mora vrijediti da je taj broj u konačnici dovoljno zamijeniti onom znamenkom koja ima taj ostatak. Sada je jasno kako izgleda niz.

Bili vi Marin ili Stjepan, potpuno je svejedno, jer kada otkrijete ovu pravilnost, dalje je lagano. Svaki blok od uzastopnih 9 elemenata zadanog intervala ima istu sumu koja iznosi  $1 + 2 + 3 + \dots + 9 = 45$ . Takvih (potpunih) blokova ima  $\lfloor \frac{r-l+1}{9} \rfloor$ . Ostalo je još pribrojiti “višak”, odnosno elemente zadnjeg, nepotpunog bloka. Tih elemenata ima  $(r - l + 1) \bmod 9$ , a prvi je jednak  $l$ . Dakle, na svaki upit možemo odgovoriti u konstantnoj složenosti.



## Zadatak: Lutrija

Predložio: Marin Kišić

Potrebno znanje: matematika, brza provjera je li broj prost, bfs/dfs

Da biste osvojili 14 bodova bilo je dovoljno najprije provjeriti je li  $|A - B|$  prost broj, ako jest, onda je niz jednosavno  $\{A, B\}$ . Ako razlika nije prosta možemo fiksirati neki  $x$  između 2 i 1000 i provjeriti je li  $x$  prost, je li  $|A - x|$  prost i je li  $|x - B|$  prost. Ako je svo troje prosto onda je rješenje niz  $\{A, x, B\}$ , inače nema rješenja.

Rješenje sljedeće parcijale je ekvivalentno rješenju cijelog zadatka samo što se za provjeru je li neki broj prost može samo provjeriti svaki broj između 2 i tog broja, a u pravom rješenju to treba napraviti u  $\mathcal{O}(\sqrt{n})$  operacija.

Konačno, za potpuno rješenje je bilo potrebno uočiti da je apsolutna razlika neka dva prosta broja veća od 2 uvijek paran broj. Kako se traži da apsolutna razlika između susjedna dva elementa niza bude prosta, možemo zaključiti da jedini brojevi koji se mogu pojaviti u nizu su  $A - 2$ ,  $A$ ,  $A + 2$ ,  $B - 2$ ,  $B$ ,  $B + 2$  i 2. Od tih sedam brojeva ostavimo samo proste. Sada, svaki broj možemo zamisliti kao čvor u grafu u kojem veza između dva čvora postoji ako je razlika pripadajućih brojeva prosta. Na kraju, bilo kojim algoritmom za traženje puta na grafu pronađemo neki put od  $A$  do  $B$ .

## Zadatak: Džumbus

Predložio: Vedran Kurdija

Potrebno znanje: ??

## Zadatak: Trobojnica

Predložio: Daniel Paleka

Potrebno znanje: ??

Kako bismo osvojili podzadatak od 10% bodova, trebamo pronaći nužan i dovoljan uvjet postojanja domoljubne triangulacije za mnogokut s obojanim stranica.

**Tvrđnja 1:** U svakoj triangulaciji za  $N \geq 4$  postoje barem dva “uha”, tj. trokut koji dijeli dvije stranice s mnogokutom.

*Skica dokaza:* Indukcija. Vrijedi za kvadrat. Neka dijagonala dijeli mnogokut na dva manja mnogokuta. Ako je neki od njih trokut, to je traženo uho; inače se možemo induktivno pozvati dalje.

Neka na početku ima  $a$ ,  $b$  i  $c$  stranica u bojama 1, 2 i 3 redom. Vrijedi  $a + b + c = N$ . Jasno je da mora biti  $\max\{a, b, c\} < n$ , inače ne može postojati uho.

**Tvrđnja 2:** Ako postoji domoljubna triangulacija, brojevi  $\{a, b, c\}$  su iste parnosti.

*Skica dokaza:* Dvostruko prebrojavanje. Prebrojimo li broj bridova boje 1 po trokutima, dobivamo:

$$2 \cdot (\text{broj dijagonala boje 1}) + a = N - 3,$$

iz čega zaključujemo da je  $a$  iste parnosti kao  $N - 3$ . Analogno dobivamo isto i za  $b$  i  $c$ .

Navedeni uvjeti su zapravo dovoljni; to se najbolje dokazuje algoritmom koji je rješenje ovog zadatka. Ukratko rečeno, induktivno ćemo raditi uši koristeći stranice dvije različite boje, čime nećemo promijeniti (dokažite!) jednakost parnosti brojeva  $\{a, b, c\}$  u novom mnogokutu, koji ima jednu stranicu manje. Jedino na što treba paziti je mogućnost da novi mnogokut ima sve stranice iste boje.

Jedna lijepa implementacija je sljedeća: za svaki vrh mnogokuta pamtimo njegovog sljedbenika u trenutnom mnogokutu, te koju brida koji ih povezuje. Traženje mjesta na kojem možemo napraviti uho zapravo nije



veliki problem. Naime, možemo samo ići u krug i napraviti uho na prvom vrhu čiji su prethodni i sljedbeni brid iste boje, te je barem jedna od tih boja ona boja koje ima najviše među trenutnim stranicama. Ukoliko svaki put krenemo od mjesta gdje smo napravili zadnje uho, moguće je pokazati da je vremenska složenost amortizirani  $\mathcal{O}(N)$ .

Postoje i kompliciranija  $\mathcal{O}(N \log N)$  rješenja, koja također trebaju dobiti sve bodove.

### **Zadatak: Zoo**

Predložio: Ivan Paljak

Potrebno znanje: ??