

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Analiza velikih skupova podataka

Autori: Goran Delač, Marin Šilić, Klemo Vladimir

Ak. god. 2016/2017

3. Laboratorijska vježba

Zadatak ove laboratorijske vježbe je programsko ostvarenje algoritma procjene broja jedinica u zadanom toku bitova koristeći pristup **Datar-Gionis-Indyk-Motwani (DGIM)**.

1.1 Algoritam DGIM

Algoritam DGIM **procjenjuje** broj jedinica zadanog toka podataka u unaprijed zadanom prozoru maksimalne veličine od **N** bitova bez čuvanja promatranog toka podataka u memoriji. Prirodni broj **N** će biti unaprijed zadan te za potrebe ove vježbe nikad neće biti veći od 10^6 , dok veličina samog ulaznog toka nula i jedinica može biti **manja ili veća** od **N**.

Algoritam koji je potrebno ostvariti oslanja se na održavanje **pretinaca** veličine jednake **potencijama broja 2** (veličina najvećeg pretinca nikad nije veća od $\log_2 N$). Dodatno, inačica algoritma koju je potrebno ostvariti u ovoj vježbi dozvoljava postojanje maksimalno **dva** pretinca iste veličine (tj. najjednostavnija inačica algoritma). Ovaj zahtjev je potrebno održavati iterativnim spajanjem blokova dok se ne zadovolji navedeni uvjet. Dodatno, pretinci se ne smiju preklapati i svaka jedinica u ulaznom toku mora biti pokrivena jednim (i samo jednim) pretincem. Konačno, potrebno je **brisati** pretince koji se nalaze izvan prozora od **N** bitova. Za tu svrhu, svaki pretinac sadrži i **vremensku oznaku** koja odgovara pojavljivanju najnovije jedinice u tom pretincu. Vremenske jedinice su pozitivni cijeli brojevi s početkom **t=0**.

Osim održavanja pretinaca potrebno je odgovarati na upite oblika **“koliko jedinica ima u zadnjih k bitova?”**, gdje je $1 \leq k \leq N$. Procjenu broja jedinica računati prema uputama iz literature - (1) pronaći najstariji pretinac **z** čija vremenska oznaka još uvijek pripada prozoru od **k**

bitova, (2) sumirati veličine svih pretinaca s recentnijim vremenskim oznakama od one pretinca z, (3) dodati sumi iz (2) pola veličine pretinca z (*zaokruženo na manji cijeli broj*).

Molimo pogledajte predavanja i literaturu za detaljno objašnjenje svih koraka algoritma:

- [AVSP_Data_Streams.pdf](#) (predavanje)
- [Maintaining stream statistics over sliding windows](#) (izvorni rad)
- [Mining Massive Datasets - Ch. 4 \(Mining Data Streams\)](#) (knjiga)

Napomena: Iako na prvi pogled algoritam DGIM može izgledati besmislen (npr. možete pomisliti da je dovoljno samo održavati jedan brojač), ključno je primjetiti da ovaj algoritam **procjenjuje** broj jedinica u zadnjih **k elemenata toka** (dakle, k je parametar upita), ali na način da u memoriji **ne sprema niti jedan jedini bit** ulaznog toka. Dodatno, zbog ograničenja SPRUT okoline za automatsko vrednovanje rješenja, nije moguće praktično simulirati **beskonačne** tokove podataka te svi zadani ispitni primjeri mogu, naravno, stati u memoriju. Međutim, ovo ograničenje ne mijenja ništa u implementaciji, tj. osnovna struktura algoritma ostaje ista.

1.2 Ulazna i izlazna datoteka

Format ulazne datoteke u program koji predajete u ovoj laboratorijskoj vježbi je:

```
N
bitovi ili upit
...
bitovi ili upit
```

Gdje je

- N - veličina prozora ($1 < N < 10^6$)
- `bitovi` - niz nula i jedinica maksimalne duljine 80 znakova u redu (zbog čitljivosti)
- `upit` - upit oblika " $q\ k$ ", gdje je k veličina prozora za upit ($1 \leq k \leq N$)

U ulaznom zapisu svaki redak (osim zadnjeg) završava znakom za kraj retka (`\n`). Linija 1. sadrži prirodni broj N . Nakon toga slijedi **proizvoljan** broj linija koje mogu sadržavati ili bitove ulaznog toka (niz nula i jedinica, bez praznina) ili upit (znak " q ", razmak, prirodni broj) na koji je potrebno odgovoriti.

Napomena: postoji samo **jedan** tok nula i jedinica u ulaznoj datoteci, bez obzira što je razlomljen na više linija i prekidan upitima! Dodatno, upiti **ne utječu** na pomicanje vremenskih oznaka pri čitanju ulaznog toka.

Primjer ulazne datoteke:

```
100
1010101101
1110101011
q 20
1000010010
q 3
```

Veličina prozora je $N = 100$. Ukupna veličina ulaznog toka jest $10 \cdot 3 = 30$ bitova. Prvi bit u toku je bit 1 ($t=0$), drugi bit je 0 ($t=1$), treći bit je 1 ($t=2$), itd. Nakon 20. bita (tj. u trenutku $t=19$) potrebno je odgovoriti na prvi upit ($q \ 20$), a nakon 30. bita na drugi upit ($q \ 3$, $t=29$). Izlaz treba sadržavati onoliko linija koliko ima upita u ulaznoj datoteci. Npr. izlaz za navedeni primjer treba biti:

```
11
0
```

Važne napomene:

- Vremensko ograničenje za izvođenje programa će biti 20 sekundi (ulazne datoteke za ispitivanje neće imati više od 10^5 linija)
- Ulazna točka za Java rješenja treba biti u razredu **DGIM**, a ulazna točka za Python rješenja treba biti u datoteci **DGIM.py**

1.3 Primjer za provjeru valjanosti

Na stranicama predmeta postavljen je primjer ulazne datoteke s pripadajućim očekivanim izlazom (*lab3_primjer.zip*). Preporučamo provjeru ispravnosti na temelju zadanog primjera prije predaje vježbe na sustav *sprut*. Evaluacija ovog zadatka provodit će se na nekoliko različitih ulaznih datoteka, uključujući i spomenuti primjer.