

Konstrukcija i analiza algoritama - pismeni ispit SEP2

23.10.2025.

Ispit traje 3h. Svaki zadatak nosi po 10 poena. Ukupan broj poena na pismenom delu ispita je 60 poena. Prag za prolaz je 17 poena.

EXACT-COVER-BY-3-SETS

Problem TAČNO-POKRIVANJE-SA-3-SKUPOVIMA (EXACT-COVER-BY-3-SETS) definiše se na sledeći način: dat je konačan skup X takav da važi $|X| = 3q$ i kolekcija $\mathcal{C}$ od podskupova skupa X koji imaju po tri elementa. Pitanje je da li $\mathcal{C}$ sadrži tačno pokrivanje skupa X , tj. da li postoji podkolekcija $\mathcal{C}' \subseteq \mathcal{C}$ takva da se svaki element iz X pojavljuje tačno u jednom članu $\mathcal{C}'$. Pošto je poznato da je problem EXACT-COVER-BY-3-SETS NP-kompletno, treba pokazati da je i problem EXACT-COVER-BY-4-SETS NP-kompletno. Pomenuta dva problema mogu se zapisati kao:

X3C. Dat je skup X takav da važi $|X| = 3q$ i kolekcija $\mathcal{C}$ podskupova od X , pri čemu svaki podskup iz $\mathcal{C}$ ima tačno 3 elementa. Da li postoji $\mathcal{C}' \subseteq \mathcal{C}$ koje tačno pokriva X (tj. svaki $x \in X$ se javlja tačno u jednom članu $\mathcal{C}'$)?

X4C. Dat je skup Y takav da je $|Y|$ deljivo sa 4 i kolekcija $\mathcal{D}$ podskupova od Y koji imaju po 4 elementa. Da li postoji $\mathcal{D}' \subseteq \mathcal{D}$ koje tačno pokriva Y ?

- **(4 poena)** Objasniti kako se X3C može svesti na Y4C (hint: Dozvoljeno je dodavanje novih elemenata skupu X).
- **(3 poena)** Dokazati da ako X ima 3-tačno pokrivanje, Y dobijen redukcijom ima 4-tačno pokrivanje.
- **(3 poena)** Dokazati da ako Y ima 4-tačno pokrivanje, X od koga je Y dobijen redukcijom ima 3-tačno pokrivanje.

Rand funkcije

- **(5 poena)** Implementirati `rand5()` koristeći `rand2()`: Data je funkcija `rand2()` koja vraća 0 ili 1 sa jednakom verovatnoćom. Potrebno je implementirati funkciju `rand5()` koja vraća ceo broj uniformno (sa istom verovatnoćom) iz skupa 0,1,2,3,4.
- **(5 poena)** Implementirati `randM()` koristeći `rand2()`: Data je funkcija `rand2()` koja vraća 0 ili 1 sa jednakom verovatnoćom. Potrebno je implementirati funkciju `randM()` koja vraća ceo broj uniformno (sa istom verovatnoćom) iz skupa 0,1,2,3,4,...,M-1.

Paralelni prefiks na mreži binarnog stabla

Na mreži računara organizovanoj kao *puno binarno stablo* sa n listova, u i -tom listu se nalazi vrednost x_i ($1 \leq i \leq n$). Neka je $*$ asocijativna binarna operacija nad skupom vrednosti ($\forall a, b, c: (a * b) * c = a * (b * c)$). Potrebno je izračunati *paralelni prefiks* (skan): nakon izvršavanja algoritma, list i treba da sadrži

$$y_i = x_1 * x_2 * \dots * x_i, \quad \text{za svaki } i = 1, \dots, n,$$

pri čemu je cilj postići vreme izvršavanja $O(\log n)$.

Pretpostavite sinhroni model: računanje teče u diskretnim rundama; u svakoj rundi svaki čvor može obraditi poruke pristigle iz dece/roditelja i poslati poruke roditelju/deci. Poruka nosi $O(1)$ elemenata domena (ili $O(1)$ mašinskih reči). Interni čvorovi nemaju sopstvene x -vrednosti (ali smeju čuvati privremene rezultate).

- **(4 poena)** Osmisliti algoritam i opisati faze
- **(3 poena)** Napraviti jednostavan primer male dimenzije i opisati kako na njemu algoritam radi
- **(3 poena)** Diskutovati složenost i korektnost algoritma

Predmeti i nastavnici

Na fakultetu postoji n predmeta i m nastavnika. Za svaki predmet znamo koji nastavnici su kvalifikovani da ga drže i za svakog nastavnika znamo koliko maksimalno predmeta on može da drži (norma nastavnika). Svaki predmet drži tačno jedan nastavnik. Ispisati da li moguće ili ne napraviti raspodelu časova tako da svi časovi budu pokriveni.

Primer ulaza:

```
3 # broj nastavnika
2 # broj predmeta
```

```

1 2 # nastavnik 1 moze da drzi dva predmeta
2 1 # nastavnik 2 moze da drzi jedan predmet
3 1 # nastavnik 3 moze da drzi jedan predmet
4 # broj kvalifikacija
1 1 # nastavnik 1 je kvalifikovan za predmet 1
1 2 # nastavnik 1 je kvalifikovan za predmet 2
2 1 # nastavnik 2 je kvalifikovan za predmet 1
3 2 # nastavnik 3 je kvalifikovan za predmet 2
Primer izlaza:
Svi casovi su pokriveni.

```

Pohlepni 3-SAT

Dopuniti fajl 2.cpp - pohlepni približni algoritam za 3-SAT tako da bude ispravan. Dopuna se odnosi na proveru da li nekoj promenljivoj možemo da promenimo valuaciju (iz 0 u 1 ili iz 1 u 0) tako da se poveća maksimalan broj trenutno zadovoljenih klauza.

```

Primer izlaza:
Approximate assignment:
x0 = true
x1 = false
x2 = false
Satisfied clauses: 2 / 2

```

Dinamična struktura

U ovom zadatku potrebno je održavati **strukturu celih brojeva** koja podržava sledeće dve osnovne operacije:

- **INSERT(S, x)** — ako broj x nije u strukturi S , umetni ga u S
- **DELETE(S, x)** — ako broj x postoji u strukturi S , izbriši ga

Pored toga, struktura mora da podržava i sledeće dve vrste upita:

- **K-TH(S)** — vrati k -ti najmanji element iz strukture S
- **COUNT(S, x)** — vrati broj elemenata u skupu S koji su strogo manji od x

Ulaz

Prva linija sadrži ceo broj Q ($1 \leq Q \leq 200\,000$) — broj operacija.

U narednih Q linija svaka linija sadrži po jednu operaciju, u sledećem formatu:

- karakter I , D , K ili C koji označava tip operacije:
 - I — umetanje (INSERT)
 - D — brisanje (DELETE)
 - K — upit za k -ti najmanji element (K-TH)
 - C — broj elemenata manjih od x (COUNT)
- zatim jedan ceo broj koji predstavlja parametar operacije

Za parametar x važi: $0 \leq |x| \leq 10^9$.

Za parametar k važi: $1 \leq k \leq 10^9$.

Izlaz

Za svaki upit K-TH i COUNT, ispisati rezultat u posebnoj liniji.

Za upit K-TH, ako je k veće od broja elemenata u skupu, ispisati reč - notvalid:

Primer ulaza:

```
8
I -1
I -1
I 2
C 0
K 2
D -1
K 1
K 2
```

Izlaz:

```
1
2
2
notvalid
```