

Konstrukcija i analiza algoritama (AIDA, KIAA)

II semestar, 2017/18

Cilj predmeta

1. **Upoznavanje** metoda dizajniranja algoritama
2. Procena efikasnosti algoritama i dokaz korektnosti algoritma
3. Obnova algoritamskih strategija-indukcija, greedy
4. Dinamičko programiranje i primena u aktuelnim informacionim tehnologijama
5. **Upoznavanje** sa novim algoritmima za rad sa grafovima
6. Algebarski problemi
7. NP kompletni problemi i približni algoritmi

Saradnja tokom kursa

jelenagr@matf.bg.ac.rs

google group ???

Raspored – run-time verzija :-)

ponedeljak 15-17 (Sveti Nikola),

petak 16-18 (718)

Konsultacije: pri ovakvom rasporedu

petak 18-19 (716), 17-18 (Zbornica,

nova zgrada u ulici Svetog Nikole 39)

Literatura

- E-priručnici sa vežbi i predavanja

<http://www.math.rs/~jelenagr/>

<http://poincare.matf.bg.ac.rs/~ezivkovm/nastava/kiaa.htm>

- 1. Živković M Algoritmika , MATF 2001.
2. CLR knjiga: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest ,Clifford Stein Introduction to Algorithms, , MIT Press, 2001.
3. Manber U Introduction to Algorithms - a creative approach, AW, 1989.

Prvi domaći zadatak

- Izneti sopstvene predloge za literaturu (ako ste već koristili neku dobru knjigu iz algoritama u ma kom programskom jeziku).
- Na sledeći čas doneti sa sobom i literaturu koju preporučujete.

Način ocenjivanja

Kolokvijum: 20 poena

Programerski zadaci i pitalice: $2 \cdot 5 = 10$ poena

Pismeni deo ispita: 30 poena

Predlozi i napomene?

Potrebna znanja i veštine

1. Programski jezik C++
 2. Diskretne strukture 1, 2
 3. Matematička analiza 1
 4. Teorija brojeva i osnovi geometrije
- Po statutu, ima/nema preduslova!!!

Koristan softver i tehnologije

Code::Blocks

Dia, GIMP

Tekst procesori i editori

Motivacija (upotrebljivost kursa)

1. Testovi za posao
2. Primeri uspešnih karijera u
istraživanju: MDCS
nauci: MIT
industriji: google.com, facebook.com

Finale

1. Predlozi?
2. Očekivanja?
3. Nejasnoće?

Zadaci za upoznavanje

1. <https://petlja.org/BubbleBee/r/2016-drzavno-oo-deljivi>

Дата су два цела броја a , b ($0 < a < b < 1\,000\,000\,000\,000\,000$) и цео број c ($1 < c < 1\,000\,000\,000\,000\,000$). Написати програм ДЕЉИВОСТ који ће исписати колико целих бројева између a и b (укључујући и a и b) је дељиво датим бројем c .

PRIMER: ULAZ 8 40 5

IZLAZ 7

```
#include <iostream>
using namespace std;
int main() {
    int a, b, k;
    cin >> a >> b >> k;
    int broj = 0;
    for (int i = a; i <= b; i++)
        if (i % k == 0)
            broj++;
    cout << broj << endl;
    return 0;
}
```

- `#include <iostream>`
`using namespace std;`

```
int main() {  
    int a, b, k;  
    cin >> a >> b >> k;  
    int l = a % k == 0 ? a/k : a/k + 1; // ceil(a/k)  
    int d = b/k; // floor(b/k)  
    int broj = d >= l ? d-l+1 : 0;  
    cout << broj << endl;  
    return 0;  
}
```

Сваке године ученици имају прилику да у школским свечаним салама присуствују прослави школске славе. Разредне старешине су одлучиле да распореде ученике према прозивнику на следећи начин: најпре се у сали попуњава први ред у смеру лево надесно, потом на исти начин се попуњава други ред и тако редом док се не попуни цела сала која има n редова са по m места за седење у сваком реду. Међутим, кад је дошао директор школе, образложио је разредним старешинама зашто такав распоред седења није коректан и предложио је прерасподелу седења на следећи начин: у сваком реду (од првог до последњег) ученици треба најпре да поуне сва прва места, потом сва друга места и тако даље. Напишите програм којим се уносе цели бројеви n , m и који израчунава и исписује колико ученика у прерасподели ће остати на својим првобитним местима. Сматрајте да школа има довољно ученика да попуни целу салу.

тест пример: ако $n = 3$ и $m = 3$, првобитни распоред ученика је

```
1 2 3
4 5 6
7 8 9
```

Након прерасподеле, распоред ученика је:

```
1 4 7
2 5 8
3 6 9
```

Ученици с редним бројевима 1, 5 и 9 ће остати на својим местима.

Најчешће грешке

У овом задатку није било потребно користити низове (ни једнодимензионе ни дводимензионе низове).

Неки решаваачи су алоцирали превише меморије декларишући матрицу димензије 30000×30000 и услед тога им програм није радио.

Такође, нека решења грубом силом нису могла да дају коректан резултат у времену од 10 секунди. Било је и решења који су добили утешне поене на тест примерима за квадратну матрицу, тј. за случај $n = m$

Анализа проблема:

Седиште се не мења ако је редни број колоне расподеле једнак редном броју колоне прерасподеле и ако аналогно важи за врсте.

```
#include <iostream>
using namespace std;
int main()
{
    long int n, m; // vrsta, kolona
    cin>>n>>m;
    long int br = 0;
    if (n==m) br=n;
    else
        for (int sediste=0;sediste<n*m;sediste++)
            if (sediste / n == sediste % m && sediste / m == sediste % n) ++br;
            //sediste se ne menja ako je redni broj kolone raspodele jednak rednom broju kolone preraspodele
            //i ako analogno vazi za vrste
    cout<<br<<endl;
    return 0;
}
```

МОЖЕ ЛИ ЕФИКАСНИЈЕ ???