

1. [20] Usvojiti efikasnu implementaciju kvadratne matrice realnih brojeva za koju se zna da je simetrična u odnosu na glavnu dijagonalu, a zatim napisati kompletne potprograme (na jeziku C ili C++) za efikasno čitanje proizvoljnog elementa, upisivanje proizvoljne vrednosti elementa i određivanje broja nenultih elemenata matrice. Obavezno komentarisati program.
2. [14] Kodirati dinamičkim *Huffman*-ovim kodovima niz simbola CEDECCBBBB ako se simboli B, C, D, E kodiraju kodovima fiksne dužine sa po dva bita 00, 01, 10 i 11 respektivno.
3. [16] U prazno B-stablo reda 3 umeću se redom ključevi 13, 23, 18, 16, 35, 30, 28, 58, 78, 88, 66, 25 a zatim se redom brišu ključevi 88, 30, 58, 16. Nacrtati izgled stabla nakon svake od navedenih izmena. Koliki je srednji broj pristupa prilikom uspešnog i neuspešnog traženja, kao i popunjenost B stabla, posle svih umetanja ključeva i u završnom stanju?
4. [16] Skicirati i objasniti efikasan algoritam za nalaženje najkraćeg rastojanja između dva zadata čvora x i y u težinskom grafu sa nenegativnim težinama, kao i za određivanje tog puta. Izvesti složenost za obe reprezentacije grafa.
5. [18] Pitanja:
 - a) Objasniti ideju interpolacionog pretraživanja i napisati izraz za određivanje indeksa elementa koji se poredi.
 - b) Šta su *Fibonacci*-jeva stabla i kakva su im glavna svojstva?
 - c) Objasniti metodu heširanja koja koristi *analizu cifara*.
 - d) Objasniti da li se primarno i sekundarno grupisanje izbegava kod dvostrukog heširanja.
6. [16] Precizno opisati sortiranje metodom umetanja sa smanjenjem inkrementa. Objasniti na čemu se zasniva efikasnost ovog metoda i kako treba birati sekvencu inkrementa . Ilustrovati rad algoritma pri sortiranju niza 78 21 53 68 27 88 70 10 57 33 u tri iteracije vodeći računa o izboru inkremenata.

Rešenja:

1. Data matrica se najjednostavnije i najefikasnije može predstaviti u vidu trougaone matrice. Pretpostavka: posmatra se gornja trougaona matrica, elementi matrice su smešteni po vrstama. Adresna funkcija je $A_{i,j} = A_{0,0} + i \cdot n + j - \frac{i \cdot (i+1)}{2}$ (jer je 0 indeks prvog element niza u C/C++). [Uporediti sa donjom trougaonom matricom. Takođe uporediti sa smeštanjem elemenata po kolonama.]. Klasa u jeziku C++ koja realizuje traženu funkcionalnost je prikazana ispod. Napomena: deo koda koji predstavlja rešenje zadatka je obojen sivo.

```
#ifndef MATRICA_H_
#define MATRICA_H_
#include <iostream>
using namespace std;
class Matrica
{
protected:
    double *m_matrica;
    unsigned int m_dim;
    unsigned int napravi(unsigned int dim);
    void kopiraj(const Matrica &m);
public:
    Matrica(unsigned int dim);
    Matrica(const Matrica &m) { kopiraj(m); }
    ~Matrica() { delete []m_matrica; }

    Matrica &operator=(const Matrica &m)
    {
        if( this != &m )
        {
            delete []m_matrica;
            kopiraj(m);
        }
        return *this;
    }
    //-----
    double & dohvati(unsigned int v, unsigned int k);
    const double & dohvati(unsigned int v, unsigned int k) const
    {
        return const_cast<Matrica *>(this)->dohvati(v,k);
    }

    unsigned int brojNenultih() const;

    friend ostream & operator<<(ostream &o, const Matrica &m);
};
#endif
```

```
#include "Matrica.h"
unsigned int Matrica::napravi(unsigned int dim)
{
    m_dim = dim;
    unsigned int duz = m_dim * (m_dim+1) / 2;
    m_matrica = new double[duz];
    return duz;
}
void Matrica::kopiraj(const Matrica &m)
{
    unsigned int duz = napravi(m.m_dim);
    for(unsigned int i = 0; i < duz; m_matrica[i] = m.m_matrica[i++]);
}
Matrica::Matrica(unsigned int dim)
{
    unsigned int duz = napravi(dim);
    for(unsigned int i = 0; i < duz; m_matrica[i++] = 0);
}
double & Matrica::dohvati(unsigned int v, unsigned int k)
{
    unsigned int vrsta, kolona;
    if( v > k ) vrsta = k, kolona = v;
    else vrsta = v, kolona = k;
    unsigned int pomeraj = vrsta*m_dim + kolona -
vrsta*(vrsta+1)/2;
    return m_matrica[pomeraj];
}
unsigned int Matrica::brojNenultih() const
{
    unsigned int broj = 0;
    for(unsigned int i = 0; i < m_dim; i++)
        for(unsigned int j = i; j < m_dim; j++)
            if( dohvati(i, j) != 0 )
                broj += i == j ? 1 : 2;
    return broj;
}
ostream & operator<<(ostream &o, const Matrica &m)
{
    for(unsigned int i = 0; i < m.m_dim; i++ )
    {
        for( unsigned j = 0; j < m.m_dim; j++ )
            o << m.dohvati(i, j) << " ";
        o << endl;
    }
    return o;
}
```

```

#include "Matrica.h"
#include <iostream>
using namespace std;

void main()
{
    Matrica mat(5);

    for(unsigned int i = 0; i < 5; i++)
        mat.dohvati(i,0) = i+1;

    cout << mat << mat.brojNenultih() << endl;
}

```

Izlaz nakon pokretanja datog programa:

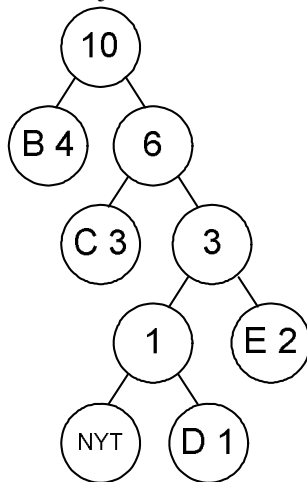
```

1 2 3 4 5
2 0 0 0 0
3 0 0 0 0
4 0 0 0 0
5 0 0 0 0
9

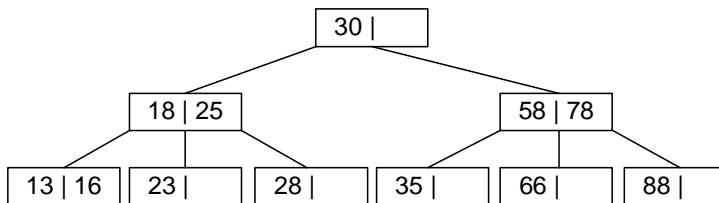
```

2. **C 0E 00D 11 11 11 100B 1001 111 10**

Izgled stabla nakon kodiranja:



3.

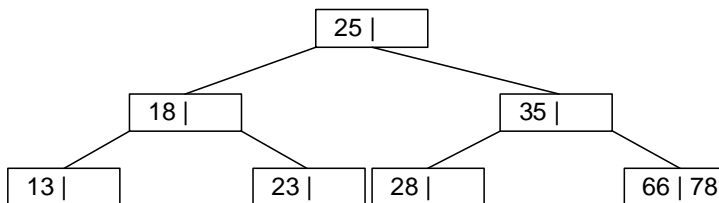


Prosečan broj pristupa:

Uspešno: $30/12 = 2.5$

Neuspešno: 3

Popunjenost: $12/18 = 2/3 \sim 0.67$



Prosečan broj pristupa:

Uspešno: $20/8 = 2.5$

Neuspešno: 3

Popunjenost: $8/14 \sim 0.57$