**SVEUČILIŠTE U RIJECI**

**FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA**

**Preddiplomski studij informatike**

**Projekti zadatak iz kolegija  
ALGORITMI I STRUKTURE PODATAKA**

**TEHNIČKA DOKUMENTACIJA**

**Autor: Mateo Trakoštanec**

# Sadržaj

Contents

[Sadržaj 1](#_Toc138285486)

[1. Uvod 2](#_Toc138285487)

[2. Odabir podataka i načina sortiranja 2](#_Toc138285488)

[3. AVL stablo 3](#_Toc138285489)

[4. Rezultati 12](#_Toc138285490)

# Uvod

Cilj ovog projekton zadatka je bio napraviti program koji će čitati podatke iz datoteke unositi ih u AVL stablo gdje će se isti sortirati. Na kraju nam se nudi odabir ispisa podataka, sortirani na željeni način. U nastavku slijedi tehnička dokumentacija ovog zadataka i način na koji sam implementirao svoje riješenje.

# Odabir podataka i načina sortiranja

Podatke koje sam koristio za ovaj projekt sam dobio dodavanjem broja 1 na ostatak kod dijeljenja broja 8042, zadnje četiri znamenke mog JMBAG broja, sa brojem 113 kako je bilo napisano uputama. Time sam dobio broj 20 i podatke iz skupa „*French financial news*“ koji su dostupni na sljedećem linku: <https://www.kaggle.com/datasets/arcticgiant/french-financial-news>. Iz tog skupa podataka nasumično sam odabrao 25 redaka koji su činili moj dataset. Atribut koji ću koristiti kao index za sortiranje je drugi po redu odnosno „*Numero page*“.

Kako bi odredio način sortiranja koristio sam isti način no ovaj puta sam 8042 dijelio sa 7 i dobio broj 7, odnosno 8042%7+1=7, AVL stablo. Pošto nigdje nije određeno kojim načinom smo morali sortirati AVL stablo, napravio sam izbornik gdje korisnik može odlučiti između 3 načina sortiranja stabla(preorder, inorder i postorder). Također sam dodao mogućnost ispisa stabla.

Iz datoteke sam učitavao sve podatke i unosio ih u polje, no zbog preglednosti ispis se vrši nad „*Numero page*“ , „*Date*“ , „*Nbr image*“ , „days to 2010“, „*Sentiment Vader Title*“ i „*Title eng*“ od kojih je „*Numero page*“ primarni ključ kojim se sortira stablo. Izradio sam novu csv datoteku sa 25 nasumično odabranih redova i promijenio sam način odvajanja podataka u csv datoteci sa ',' na ';' jer su neki stupci sadržavali tekst koji je koristio zareze te samim time se krivo učitavalo u polje.

Podaci koju su bili korišteni kao dataset:

A screenshot of a computer

Description automatically generated with medium confidence

# AVL stablo

AVL stablo je oblik binarnog stabla koje je dodatno izbalansirano, što znači da je visina lijevog i desnog podstabla svakog čvora uravnotežena ili razlikuje najviše za jedan. Za izradu AVL stabla su nam potrebne funkcije za rotaciju stabla i funkcija za izračun visine stabla.

Postoje 4 vrste rotacija stabla:

1. Lijeva rotacija

*void* LeftRotation(Node \*& *root*) //lijeva rotacija

{

    Node\* temp = root->left;

    root->left = temp->right;

    temp->right = root;

    root = temp;

}

1. Desna rotacija

*void* RightRotation(Node\*& *root*) //desno rotacija

{

    Node\* temp = root->right;

    root->right = temp->left;

    temp->left = root;

    root = temp;

}

1. Dupla lijeva rotacija

*void* DoubleLR(Node \*& *root*) //dupla lijeva rotacija

{

    RightRotation(root->left);

    LeftRotation(root);

}

1. Dupla desna rotacija

*void* DoubleRR(Node \*& *root*) //dupla desna rotacija

{

    LeftRotation(root->right);

    RightRotation(root);

}

Funkcija za izračun visine stabla:

*int* height(Node\* *root*) //izracun visine cvora

{

    if(root==NULL) return -1;

    else

    {

*int* LHeight = height(root->left);

*int* DHeight = height(root->right);

        if(LHeight > DHeight) return LHeight+1;

        else return DHeight+1;

    }

}

Kako bih unjeli podatke iz datoteke u AVL stablo koristimo sljedeću funkciju:

Node\* addAVL(Node\*& *root*, French *x*) // dodavanje odabranog podatka num2 u AVL stablo

{

    if(root == NULL)

        root = new Node(x);

    else if(x.num2 < root->data.num2)

    {

        addAVL(root->left, x);

        if(height(root->left)-height(root->right)==2)

        {

            if(x.num2 < root->left->data.num2)

                LeftRotation(root);

            else DoubleLR(root);

        }

    }

    else if(x.num2 > root->data.num2)

    {

        addAVL(root->right, x);

        if(height(root->right)-height(root->left)==2)

        {

            if(x.num2 > root->right->data.num2)

                RightRotation(root);

            else DoubleRR(root);

        }

    }

    return root;

}

Ta funkcija se poziva nakon što iz datoteke učitamo sve podatke u varijable

    vector<vector<string>> parsedCSV; // stupac

    vector<string> parsedRow; // red

    string line, cell;

    fstream file("dataset.csv", ios::in); // otvaranje datoteke

    if (file.is\_open())

    {

        while (getline(file, line)) // podjela csv doc na redke i stupce

        {

            parsedRow.clear();

            stringstream str(line);

            while (getline(str, cell, ';'))

                parsedRow.push\_back(cell);

            parsedCSV.push\_back(parsedRow);

        }

    }

else

    {

        cout << "ERROR: Failed to open the file." << endl;

    }

Te ih pomoću for petlje unosimo u array koji je napravljen pomoću strukture te sadrži sve potrebne varijable koje koristimo

*struct* French // struktura kojom izradujemo polje kako bi unijeli podatke iz csv

{

*int* num1, num2, num3, nbrImg;

    string date1, time, date2, titleFr, contentFr, agency, URL, textURLFr, titleEng, contentEng, textURLEng;

*float* secTo2010, daysTo2010, SVTitle, SVText, SVTextURL;

};

*struct* French arr[25]; // deklaracija polja pomocu structure u main-u

for (*int* i = 1; i <= 25; i++) // unos iz stupaca u polje

    {

        arr[i - 1].num1 = stoi(parsedCSV[i][0]);

        arr[i - 1].num2 = stoi(parsedCSV[i][1]);

        arr[i - 1].num3 = stoi(parsedCSV[i][2]);

        arr[i - 1].date1 = parsedCSV[i][3];

        arr[i - 1].time = parsedCSV[i][4];

        arr[i - 1].titleFr = parsedCSV[i][5];

        arr[i - 1].contentFr = parsedCSV[i][6];

        arr[i - 1].agency= parsedCSV[i][7];

        arr[i - 1].URL= parsedCSV[i][8];

        arr[i - 1].textURLFr= parsedCSV[i][9];

        arr[i - 1].nbrImg = stoi(parsedCSV[i][10]);

        arr[i - 1].secTo2010 = stof(parsedCSV[i][11]);

        arr[i - 1].daysTo2010 = stof(parsedCSV[i][12]);

        arr[i - 1].date2 = parsedCSV[i][13];

        arr[i - 1].titleEng = parsedCSV[i][14];

        arr[i - 1].contentEng = parsedCSV[i][15];

        arr[i - 1].textURLEng= parsedCSV[i][16];

        arr[i - 1].SVTitle = stof(parsedCSV[i][17]);

        arr[i - 1].SVText = stof(parsedCSV[i][18]);

        arr[i - 1].SVTextURL = stof(parsedCSV[i][19]);

    }

Pozivom funkcije u čvor se unose podaci iz varijable num2 te se automatski sortiraju u AVL stablo

*struct* Node // kreiranje strukture cvora

{

    French data;

    Node\* left;

    Node\* right;

    Node (French da)

    {

        data.num1 = da.num1;

        data.num2 = da.num2;

        data.num3 = da.num3;

        data.date1 = da.date1;

        data.time = da.time;

        data.titleFr = da.titleFr;

        data.contentFr = da.contentFr;

        data.agency = da.agency;

        data.URL = da.URL;

        data.textURLFr = da.textURLFr;

        data.nbrImg = da.nbrImg;

        data.secTo2010 = da.secTo2010;

        data.daysTo2010 = da.daysTo2010;

        data.date2 = da.date2;

        data.titleEng = da.titleEng;

        data.contentEng = da.contentEng;

        data.textURLEng = da.textURLEng;

        data.SVTitle = da.SVTitle;

        data.SVText = da.SVText;

        data.SVTextURL = da.SVTextURL;

        left = NULL;

        right = NULL;

    }

};

Node\* root = new Node(arr[0]); // kreiranje glavnog cvora

    for(*int* i = 1; i<=24; i++) // unos podataka polja u AVL stablo pomocu funkcije

    {

        addAVL(root, arr[i]);

    }

Ispis stabla se vrši uz pomoć izbornika koji poziva funkcije ovisno o odabiru korisnika.

 cout<<"Ispis sortiranih podataka pomocu AVL stabla sa indexom \"Numero Page\""<<endl;

    cout<<"U ISPISU SU UREDENI I MAKNUTI PODACI RADI BOLJE PREGLEDNOSTI"<<endl<<endl;

*int* odabir=10;

    while(odabir!=0){

        cout<<"========================"<<endl;

        cout<<"1. Ispis podataka PreOrder"<<endl;

        cout<<"2. Ispis podataka InOrder"<<endl;

        cout<<"3. Ispis podataka PostOrder"<<endl;

        cout<<"4. Ispis stabla"<<endl;

        cout<<"0. Izlaz iz programa"<<endl;

        cout<<"========================"<<endl;

        cout<<"Odaberite nacin ispisa: ";

        cin>>odabir;

        cout<<endl;

        switch(odabir){

        case 1:

            cout<<"Odabrali ste ispis PreOrder redosljedom!"<<endl;

            cout<<setw(5)<<"Numero page |     Date     | Nbr image | days to 2010 |  SVTitle  | Title eng  "<<endl;

            PrintTreePreOrder(root);

            break;

        case 2:

            cout<<"Odabrali ste ispis InOrder redosljedom!"<<endl;

            cout<<setw(5)<<"Numero page |     Date     | Nbr image | days to 2010 |  SVTitle  | Title eng  "<<endl;

            PrintTreeInOrder(root);

            break;

        case 3:

            cout<<"Odabrali ste ispis PostOrder redosljedom!"<<endl;

            cout<<setw(5)<<"Numero page |     Date     | Nbr image | days to 2010 |  SVTitle  | Title eng  "<<endl;

            PrintTreePostOrder(root);

            break;

        case 4:

            cout<<"Odabrali ste ispis stabla!"<<endl;

            printTree(root,5);

            break;

                  case 0: break;

        default:

            cout<<"Neispravan unos, pokusajte ponovno!"<<endl;

        }

    }

    file.close();

    return 0;

}

Funcije za ispis stabla su sljedeće:

1. Preorder

*void* PrintTreePreOrder(Node \**root*) //preOrder ispis

{

    if (root==0)

    {

        return;

    }

    cout<<setw(5)<<root->data.num2<<"          ";

    cout<<setw(5)<<root->data.date1<<"     ";

    cout<<setw(5)<<root->data.nbrImg<<"         ";

    cout<<setw(5)<<root->data.daysTo2010<<"     ";

    cout<<setw(5)<<root->data.SVTitle<<"      ";

    cout<<setw(5)<<root->data.titleEng;

    cout<<endl;

    PrintTreePreOrder(root->left);

    PrintTreePreOrder(root->right);

}

1. Inorder

*void* PrintTreeInOrder(Node \**root*) //inOrder ispis

{

    if (root==0)

    {

        return;

    }

    PrintTreeInOrder(root->left);

    cout<<setw(5)<<root->data.num2<<"          ";

    cout<<setw(5)<<root->data.date1<<"     ";

    cout<<setw(5)<<root->data.nbrImg<<"         ";

    cout<<setw(5)<<root->data.daysTo2010<<"     ";

    cout<<setw(5)<<root->data.SVTitle<<"      ";

    cout<<setw(5)<<root->data.titleEng;

    cout<<endl;

    PrintTreeInOrder(root->right);

}

1. Postorder

*void* PrintTreePostOrder(Node \**root*) //postOrder ispis

{

    if (root==0)

    {

        return;

    }

    PrintTreePostOrder(root->left);

    PrintTreePostOrder(root->right);

    cout<<setw(5)<<root->data.num2<<"          ";

    cout<<setw(5)<<root->data.date1<<"     ";

    cout<<setw(5)<<root->data.nbrImg<<"         ";

    cout<<setw(5)<<root->data.daysTo2010<<"     ";

    cout<<setw(5)<<root->data.SVTitle<<"      ";

    cout<<setw(5)<<root->data.titleEng;

    cout<<endl;

}

1. Ispis stabla

*void* printTree(Node \**root*, *int* *sirina*) //ispis stabla

{

    if (root==0)

    {

        cout<<setw(sirina)<<"\*"<<endl;

        return;

    }

    printTree(root->right, sirina+6);

    cout << setw(sirina);

    cout << root->data.num2 << "<" << endl;

    printTree(root->left, sirina+6);

}

Još valja napomenuti da se tijekom ispisa koriste predefinirani razmaci kako bi dizajn ispisa bio pregledan.

Korištene su sljedeće biblioteke kako bi nam omogućile čitanje datoteka, rad s vektorima i stiliziranje dokumenta:

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <iomanip>

# Rezultati

1. Ispis preorderom

A picture containing text, screenshot, menu, font

Description automatically generated

1. Ispis inorderm

A picture containing text, screenshot, menu, font

Description automatically generated

1. Ispis postorderom

A black screen with white text

Description automatically generated with low confidence

1. Ispis stable

A picture containing text, screenshot, design

Description automatically generated