****

**SEMINARSKI RAD**

Vizuelizacija kompresije podataka bez gubitaka putem metoda: Shanon-Fano i Huffman

MULTIMEDIJALNI SISTEMI

**Tim:**

* Kenan Hadžirović
* Haris Halilović
* Dženita Ljevo
* Mirza Vučijak

**Mentori:**

* Red. Prof. Dr. Haris Šupić
* Kenan Ekinović,
* Orhan Ljubunčić

**Sarajevo, decembar, 2018.**

**Opis zadatka**

Algoritmi za kompresiju podataka mogu se efikasno izvršavati računarski ali i ručno. Obzirom da je ručno izvršavanje algoritma nepraktično za poruke duže od nekoliko desetina karaktera ili samo nekoliko različitih karaktera, oni se izvršavaju ručno samo u edukativne svrhe. Pomoćni alati – tabele i grafovi koji se crtaju prilikom ručnog izvršavanja na papiru se ne moraju upotrebljavati u kreiranju algoritma za kompjutersko izvršavanje. Prilagođavanje algoritama tim metodama zauzima više memorije i troši više procesorskog vremena. Zadatak ovog seminarskog rada jer uključiti grafički prikaz podataka u algoritme koji se izvršavaju na računaru. Nakon što se poruka komprimira, potrebno je iscrtati binarno stablo kompresije koje odgovara unesenim podacima. Na osnovu tog binarnog stabla moguće je isčitati kodove svakog unesenog znaka, analizirati strukturu, kao i frekvenciju pojavljivanja znakova u unesenoj poruci. Konačno, moguće je izračunati stepen kompresije i na taj način utvrditi efikasnost kompresije.

**IZJAVA O DOPRINOSU ČLANOVA TIMA**

**PRI IZRADI SEMINARSKOG RADA IZ PREDMETA MMS**

U tabeli ispod navesti imena svih članova tima, te aktivnosti u kojima je učestvovao svaki pojedini član tima. Pri navođenju aktivnosti za svakog člana tima koristite bilo koju kombinaciju sljedećih opcija:

* Analiza problema semenarskog rada;
* Osmišljavanje rješenja;
* Praktična implementacija seminarskog rada;
* Pisanje teksta seminarskog rada;
* Ostalo (navesti koje su to ostale eventualne aktivnosti).

|  |  |
| --- | --- |
| **Prezime i ime člana tima** | **Aktivnosti** |
| Hadžirović Kenan | * Osmišljavanje rješenja; * Praktična implementacija seminarskog rada; * Pisanje teksta seminarskog rada; * Ostalo (navesti koje su to ostale eventualne aktivnosti)   1. Prijava teme i slanje seminarskog rada |
| Halilović Haris | * Praktična implementacija seminarskog rada; * Pisanje teksta seminarskog rada; |
| Ljevo Dženita | * Analiza problema semenarskog rada; * Osmišljavanje rješenja; * Praktična implementacija seminarskog rada; * Pisanje teksta seminarskog rada; |
| Vučijak Mirza | * Analiza problema semenarskog rada; * Osmišljavanje rješenja; * Praktična implementacija seminarskog rada; |

Sadržaj

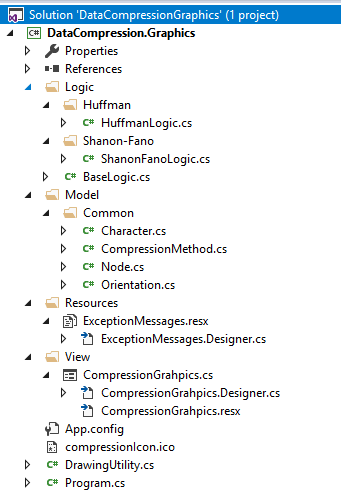
[Uvod 5](#_Toc532057986)

[Razrada 6](#_Toc532057987)

[Zaključak 7](#_Toc532057988)

[Literatura 8](#_Toc532057989)

# Uvod



# Kompresija podataka

## Shanon-Fano metoda

public static void Compress(ref List<Node> nodes, List<Character> characters)

{

Node rootNode = new Node() { IsRoot = true };

Compress(ref nodes, ref rootNode, characters, String.Empty, Orientation.Root);

}

private static void Compress(ref List<Node> nodes, ref Node parentNode, List<Character> characters, string code, Orientation orientation)

{

Node node = new Node()

{

Char = '\\',

Code = code,

LeftChild = null,

RightChild = null

};

AssignNode(parentNode, node, orientation);

nodes.Add(node);

if (characters.Count == 1)

{

node.Char = characters[0].Char;

return;

}

int splittingIndex = SplitArray(characters);

Compress(ref nodes, ref node, characters.Take(splittingIndex).ToList(), $"{code}0", Orientation.Left);

Compress(ref nodes, ref node, characters.Skip(splittingIndex).ToList(), $"{code}1", Orientation.Right);

}

private static int SplitArray(List<Character> characters)

{

int index = 1;

double upperArrayProbability = characters[0].Probability;

double lowerArrayProbability = characters.Skip(index).Sum(x => x.Probability);

double difference = Math.Abs(upperArrayProbability - lowerArrayProbability);

for (; index < characters.Count; index++)

{

upperArrayProbability += characters[index].Probability;

lowerArrayProbability -= characters[index].Probability;

double newDifference = Math.Abs(upperArrayProbability - lowerArrayProbability);

if (newDifference > difference)

{

return index;

}

else

{

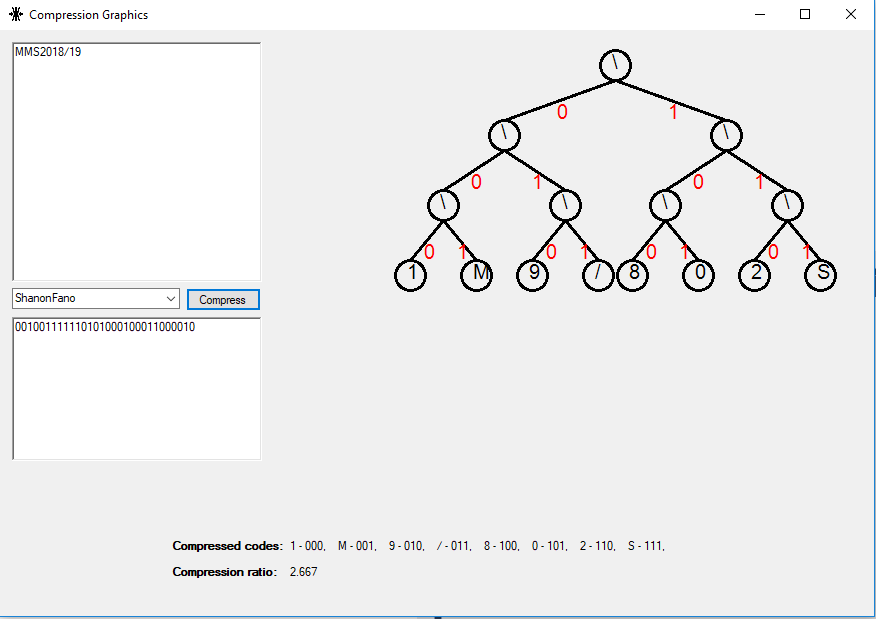
difference = newDifference;

}

}

return index;

}



Slika 1 - Prikaz binarnog stabla za Shanon-Fano algoritam

## Huffman metoda

public static void Compress(ref List<Node> nodes, List<Character> characters)

{

List<Node> workingNodes = characters.Select(x => x.MapToNode()).ToList();

while (workingNodes.Count != 1)

{

workingNodes = workingNodes.OrderBy(x => x.Probability).ToList();

workingNodes.Add(MergeNodes(workingNodes[0], workingNodes[1]));

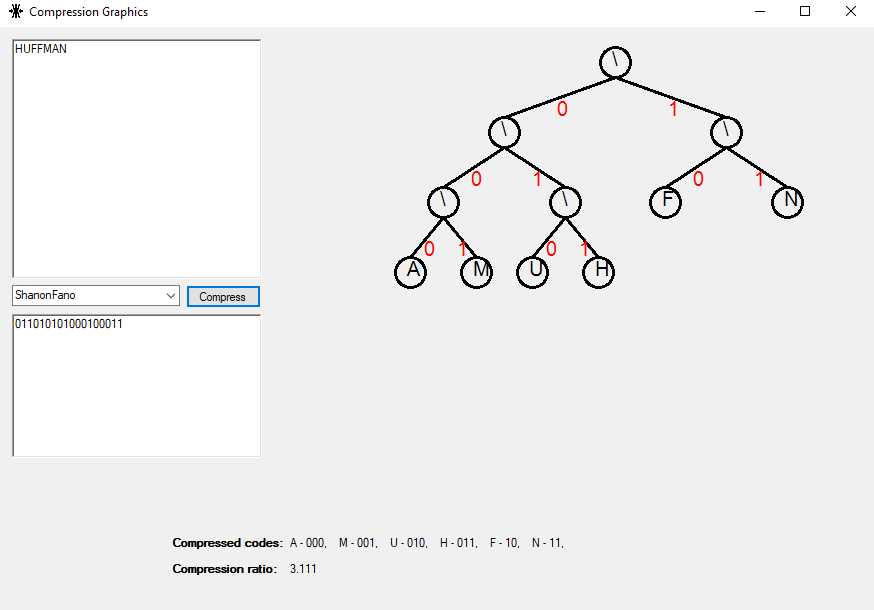
workingNodes.RemoveRange(0,2);

}

AssignCodes(workingNodes[0], String.Empty);

ExtractNodes(ref nodes, workingNodes[0]);

}



Slika 2 - Prikaz binarnog stabla za Huffman

# Zaključak

# Literatura

* Predavanja za predmet Multimedijalni Sistemi, akademska 2018/2019, H. Šupić
* Predavanja za predmet Diskretna Matematika , akademska 2018/2019 Ž. Jurić
* C# 6.0 in a Nutshell: The Definitive Reference, Joseph Albahari, Ben Albahari, O’Reilly Media, Inc

# Popis slika

Slika 1 - Prikaz binarnog stabla za Shanon-Fano algoritam 8

Slika 2 - Prikaz binarnog stabla za Huffman 9