KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Programavimo kalbų teorija (P175B124)

Laboratorinių darbų ataskaita

Atliko:

IFF-1/8 gr. studentas

Matas Palujanskas

2023 m. kovo 1 d.

Priėmė:

doc. Sajavičius Svajūnas

KAUNAS 2023

TURINYS

1. C++(L1) 3

1.1. Darbo užduotis 3

1.2. Programos tekstas 4

1.3. Pradiniai duomenys ir rezultatai 7

# C++(L1)

## Darbo užduotis

p575

Paveikslėlis, kuriame yra žinutė

Automatiškai sugeneruotas aprašymas

Paveikslėlis, kuriame yra žinutė

Automatiškai sugeneruotas aprašymas

## Programos tekstas

#include <iostream>

#include <fstream>

#include <string>

#include <cmath>

#include <chrono>

using namespace std;

using namespace std::chrono;

/// <summary>

/// Skew binary to decimal number class

/// </summary>

struct SkewBinaryToDecimal

{

string line; //binary number

/// <summary>

/// Converting from skew binary to decimal number

/// </summary>

/// <param name="line"> Skew binary number </param>

/// <param name="n"> Amount of numbers </param>

/// <returns> Converted decimal number </returns>

int to\_decimal(string line, int& n)

{

int res = 0;

for (int j = 0; j < n; j++)

{

int l = line.size();

for (int i = 0; i < l; i++) {

res += (((int)line[i] - 48) \* (pow(2, l - i) - 1));

}

return res;

}

}

};

/// <summary>

/// Decimal to skew binary number class

/// </summary>

struct DecimalToSkewBinary {

int decimal; //decimal number

string decimalToSkewBinary(int number) {

if (number == 0) {

return "0";

}

string result = "";

while (number > 0) {

int remainder = number % 3;

result = to\_string(remainder) + result;

number /= 3;

if (remainder == 2 && number > 0) {

number++;

}

}

return result;

}

};

/// <summary>

/// Printing converted numbers to file

/// </summary>

/// <param name="outputFile"> Output file </param>

/// <param name="line"> Binary numbers </param>

/// <param name="convertedNumber"> Decimal numbers </param>

/// <param name="append"></param>

void PrintResult1(string outputFile, string line,

int convertedNumber, bool append)

{

ofstream result;

if (append)

result.open(outputFile, ios\_base::app);

else

result.open(outputFile);

result << "Skew binary: " << line << " " << "decimal: " << convertedNumber << endl;

return;

}

/// <summary>

/// Reading data and performing all tasks

/// </summary>

/// <param name="inputFile"> Data file </param>

/// <param name="outputFile"> Result file </param>

void ReadAndPerformToDecimal(string inputFile, string outputFile)

{

int n;// Amount of numbers

SkewBinaryToDecimal toDecimal[20];

ifstream data(inputFile);

data >> n;

for (int i = 0; i < n; i++)

{

string line;

data.ignore();

data >> toDecimal[i].line;

line = toDecimal[i].line;

int calculated = toDecimal->to\_decimal(line, n);

//string skew\_Binary = toBinary->to\_skew\_binary(calculated, n)

bool append = false;

if (i > 0)

append = true;

PrintResult1(outputFile, line, calculated, append);

}

data.close();

return;

}

// <summary>

/// Printing converted numbers to file

/// </summary>

/// <param name="outputFile"> Output file </param>

/// <param name="line"> Binary numbers </param>

/// <param name="convertedNumber"> Decimal numbers </param>

/// <param name="append"></param>

void PrintResult2(string outputFile, int line,

string convertedNumber, bool append)

{

ofstream result;

if (append)

result.open(outputFile, ios\_base::app);

else

result.open(outputFile);

result << "Decimal: " << line << " " << "skew binary: " << convertedNumber << endl;

return;

}

/// <summary>

/// Reading data and performing all tasks

/// </summary>

/// <param name="inputFile"> Data file </param>

/// <param name="outputFile"> Result file </param>

void ReadAndPerformToBinary(string inputFile, string outputFile)

{

int n;// Amount of numbers

DecimalToSkewBinary toBinary[100];

ifstream data(inputFile);

data >> n;

for (int i = 0; i < n; i++)

{

int line;

data.ignore();

data >> line;

//line = toBinary[i].decimal;

string calculated = toBinary->decimalToSkewBinary(line);

bool append = false;

if (i > 0)

append = true;

PrintResult2(outputFile, line, calculated, append);

}

data.close();

return;

}

int main()

{

string inputFile1 = "Data1.txt";

string outputFile1 = "Results1.txt";

string inputFile2 = "Data2.txt";

string outputFile2 = "Results2.txt";

// Duration of operations start point

auto start = high\_resolution\_clock::now();

// Main calculations method

ReadAndPerformToDecimal(inputFile1, outputFile1);

ReadAndPerformToBinary(inputFile2, outputFile2);

// Duration of operations end point

auto stop = high\_resolution\_clock::now();

// Duration

auto duration = duration\_cast<microseconds>(stop - start);

cout << "Time taken by function: "

<< duration.count() << " microseconds" << endl;

return 0;

}

## Pradiniai duomenys ir rezultatai

Data1( skew binary numbers)

Data1.txt:

8

10120

200000000000000000000000000000

10

1000000000000000000000000000000

11

100

11111000001110000101101102000

0

Result1.txt:

Skew binary: 10120 decimal: 44

Skew binary: 200000000000000000000000000000 decimal: 2147483646

Skew binary: 10 decimal: 3

Skew binary: 1000000000000000000000000000000 decimal: 2147483647

Skew binary: 11 decimal: 4

Skew binary: 100 decimal: 7

Skew binary: 11111000001110000101101102000 decimal: 1041110737

Skew binary: 0 decimal: 0

Data2 ( binary numbers)

Data2.txt:

8

44

2147483646

3

2147483647

4

7

1041110737

0

Result2.txt:

Decimal: 44 skew binary: 2202

Decimal: 2147483646 skew binary: 20220200022111212100

Decimal: 3 skew binary: 10

Decimal: 2147483647 skew binary: 20220200022111212101

Decimal: 4 skew binary: 11

Decimal: 7 skew binary: 21

Decimal: 1041110737 skew binary: 10201220001020110021

Decimal: 0 skew binary: 0