МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Канонические коды деревьев

АВТОРЕФЕРАТ

дипломной работы

студентки 6 курса 631 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Багровой Кристины Андреевны

Научный руководитель

д. ф.-м. н., доцент ____________ М. Б. Абросимов

21.01.2023 г.

Заведующий кафедрой

д. ф.-м. н., доцент ____________ М. Б. Абросимов

21.01.2023 г.

Саратов 2023

ВВЕДЕНИЕ

Важным шагом в формировании теории графов явилось выделение отдельных классов графов в целях их математического исследования. Одним из важнейших классов графов являются деревья. Они широко используются в теории электрических цепей, химии, вычислительной технике и в информатике. Многие задачи, весьма сложные в общем случае, эффективно решаются для деревьев.

Имея в руках канонические коды – уникальную строку, которая не зависит от порядка нумерации вершин, можно определять изоморфизм графов простым сравнением строк, не делая дорогостоящей процедуры проверки изоморфизма. Например, как найти дубликаты в множестве N деревьев? Вычислить канонические коды, отсортировать их и последовательно сравнить. Это гораздо выгоднее, чем делать $\frac{N(N-1)}{2}$ проверок изоморфизма.

В данной работе рассматриваются различные канонические коды деревьев и генерация неизоморфных свободных деревьев с их помощью. Также в практической части будет реализована программа для анализа деревьев, вычисления различных кодов деревьев и генерации неизоморфных свободных деревьев с поддержкой нескольких способов визуализации деревьев и различных форматов входных данных.

Целью данной работы является изучение канонических кодов деревьев, их построение и генерация. Для достижения поставленных целей необходимо было разработать программный комплекс с графическим интерфейсом для работы с деревьями, позволяющий вычислять и генерировать канонические коды.

Дипломная работа состоит из введения, 8 разделов, заключения, списка использованных источников и 3 приложений. Общий объем работы – 76 страниц, из них 36 страниц – основное содержание, включая 19 рисунков и 1 таблицу, список использованных источников из 18 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

В первом разделе приводятся теоретические понятия, необходимые для дальнейшего рассмотрения темы. Здесь излагаются основные определения и обозначения, относящиеся к теории графов, а также примеры, наглядно демонстрирующие предмет изучения.

Неориентированным графом (или, для краткости, *графом*) называется пара $G = (V, \alpha)$, где α – симметричное и антирефлексивное отношение на множестве вершин V.

Расстояние от вершины u графа $G = (V, \alpha)$ до наиболее удаленной от нее вершины называется *эксцентриситетом* $e(u)$ вершины u:

$$e(u) = max_{v \in V} d(u, v). \tag{1}$$

Наименьший из эксцентриситетов вершин называется *радиусом* $r(G)$ графа G, а наибольший – *диаметром* $d(G)$:

$$r(G) = min_{u \in V}\, e(u) = min_{u \in V} max_{v \in V}\, d(u, v), \tag{2}$$

$$d(G) = max_{u \in V}\, e(u) = max_{u \in V} max_{v \in V}\, d(u, v). \tag{3}$$

Вершина, эксцентриситет которой равен радиусу, называется *центральной*. *Центр* графа – это множество его центральных вершин.

Дерево – это связный граф, в котором нет циклов.

Дерево называется *корневым*, если в нем выделена вершина r, именуемая корнем. Некорневое дерево иногда называют *свободным*.

Канонический код – способ записи графа, инвариантный относительно изоморфизма.

Во втором разделе рассматривается теорема о дереве и построение его плоского изображения.

Теорема 1. Граф G тогда и только тогда является деревом, когда в нем выполняется одно из следующих условий:

1) любые две вершины графа G соединены единственной цепью;

2) G – связный граф и число вершин в нем на единицу больше числа ребер: $n = m + 1$;

3) G не содержит циклов и $n = m + 1$.

В третьем разделе рассматриваются различные способы визуализации деревьев: поуровневая укладка, радиальная поуровневая укладка и hv-изображения.

На рисунке 1 показан пример поуровневого изображения дерева.

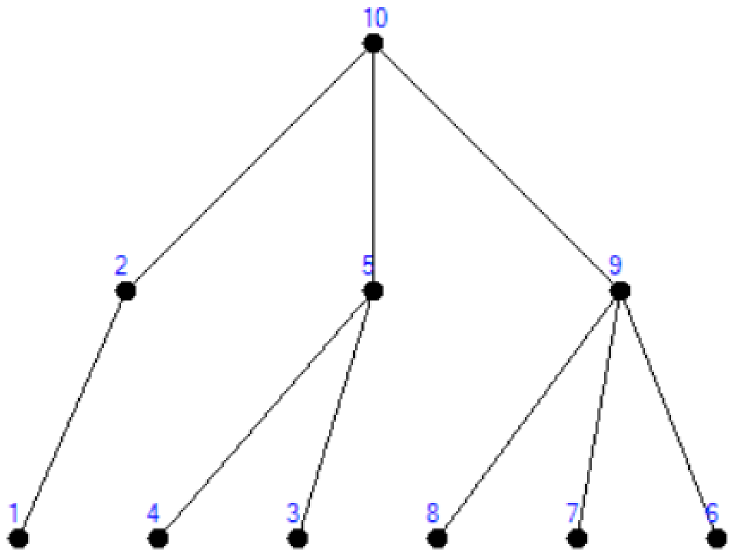


Рисунок 1 – Поуровневая укладка дерева

Для радиальной поуровневой укладки выбор угла β_q секторного сегмента W_p для поддерева с корнем p определяется с учетом числа вершин $l(p)$ в поддереве следующим образом. Пусть p лежит на уровне C_i, тогда для каждого сына q вершины p имеем

$$\beta_q = \min\left(\frac{l(q)\beta_p}{l(p)}, \tau\right), \tag{4}$$

где τ – это угол области F_p, определяемый точками пересечения a и b уровня C_{i+1} и касатаельной к уровню C_i, проведенной через точку p.

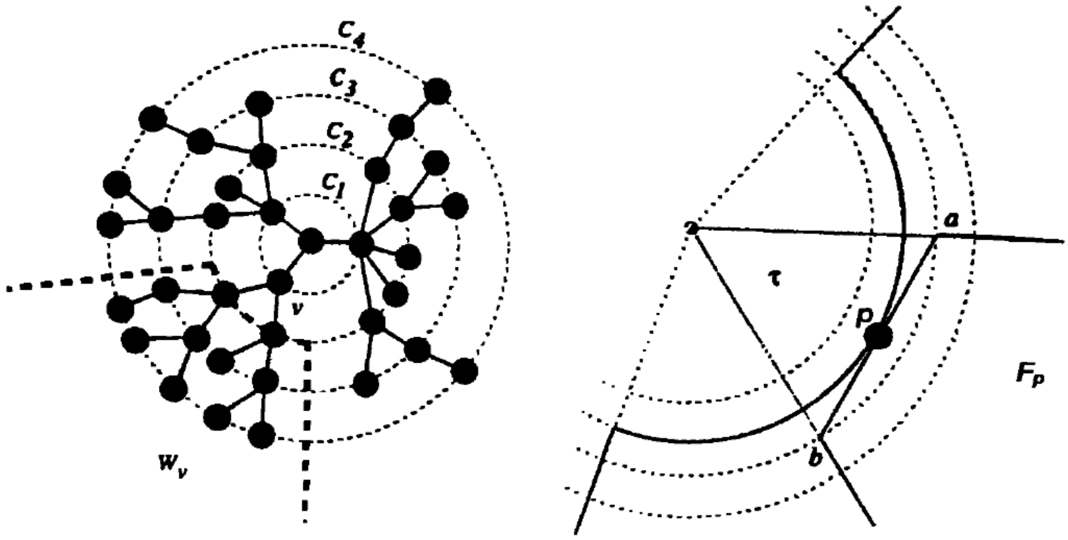


Рисунок 2 – Радиальная поуровневая укладка дерева

На рисунке 2 показаны пример радиального изображения графа и определение области размещения.

Другой вид представлений деревьев – это так называемые hv-изображения (horizontal-vertical drawing). Правила построения показаны на рисунке 3.

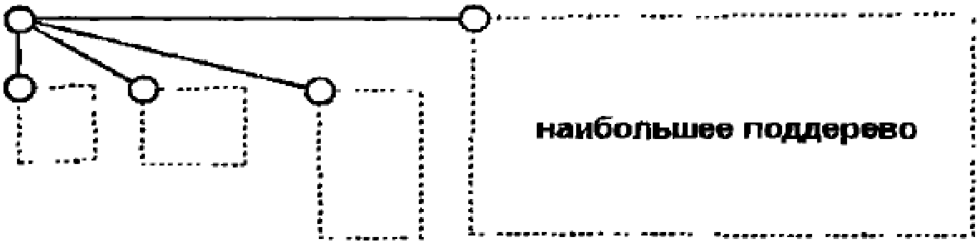


Рисунок 3 – Правила построения hv-изображения корневого дерева

В четвертом разделе рассматривается задача проверки деревьев на изоморфизм и способы её решения, одним из которых являются канонические коды, рассмотренные в следующем разделе.

Определение. Два неупорядоченных дерева $T_1 = (V_1, \alpha_1)$ и $T_2 = (V_2, \alpha_2)$ изоморфны, обозначим через $T_1 \cong T_2$, если существует биекция $M \subseteq V_1 \times V_2$ такая, что, $(root[T_1], root[T_2]) \in M$ и выполняется следующее условие $parent[v] = parent[w]$ для всех некорневых $v \in V_1$ и $w \in V_2$, $(v, w) \in M$. В таком случае M является изоморфизмом неупорядоченных деревьев или просто изоморфизмом T_1 в T_2.

Известно несколько алгоритмов проверки изоморфизма деревьев. Задача таких алгоритмов – сравнение структуры деревьев без учета меток вершин. Например, в алгоритме Ахо-Хопкрофта-Ульмана узлам деревьев приписываются целые числа, начиная с листьев – узлы уровня 0 и двигаясь вверх до корней. Деревья изоморфны тогда и только тогда, когда их корням приписано одно и то же число.

Также задачу проверки на изоморфизм двух неупорядоченных деревьев можно решить с помощью канонических кодов.

В пятом разделе рассматриваются некоторые коды деревьев: канонический уровневый код, канонический WAV-код, код Прюфера, канонический двоичный код и канонический WS-код; их алгоритмы построения и распаковки.

Уровневый код – это последовательность расстояний вершин от корня, записанных в постфиксном порядке. Каноническим называется наименьший в лексикографическом порядке среди всех уровневый код.

Для свободного дерева строится главный канонический уровневый код, которым является канонический уровневый код для дерева с центром дерева в качестве корня. Если дерево бицентральное, выбирается лексикографически наименьший, например, как на рисунке 4 справа. Главный канонический уровневый код этого дерева имеет вид:

$$L^*(T, z_1) = (3, 2, 3, 3, 2, 3, 4, 4, 3, 2, 1). \quad (5)$$

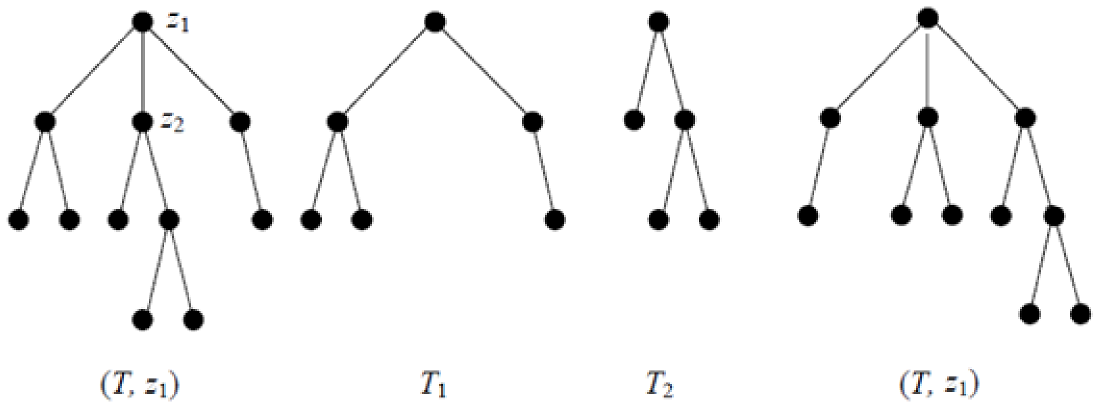


Рисунок 4 – Главный канонический уровневый код дерева (T, z).

При построении главного канонического уровневого кода требуется центр дерева, для этого используется теорема о центре дерева и алгоритм поиска центра дерева на основе этой теоремы.

Теорема 2. Каждое дерево имеет центр, состоящий или из одной вершины, или из двух смежных вершин.

Алгоритм нахождения центра дерева описан в доказательстве теоремы: все листья удаляются, пока не останется 1 или 2 вершины – центр дерева.

WAV-код (*the walk-around valency code*) состоит из последовательности чисел – количества детей каждой вершины в определенном порядке. Сначала даются теги листьям, потом смежным с ними вершинам, а листья удаляются. И так происходит, пока не останется 1 или 2 вершины. Пример построения на рисунке 5, WAV-код: 4, 3, 0, 2, 0, 1, 0, 3, 0, 0, 0, 0, 3, 0, 0, 0, 2, 2, 0, 0, 2, 0, 0.

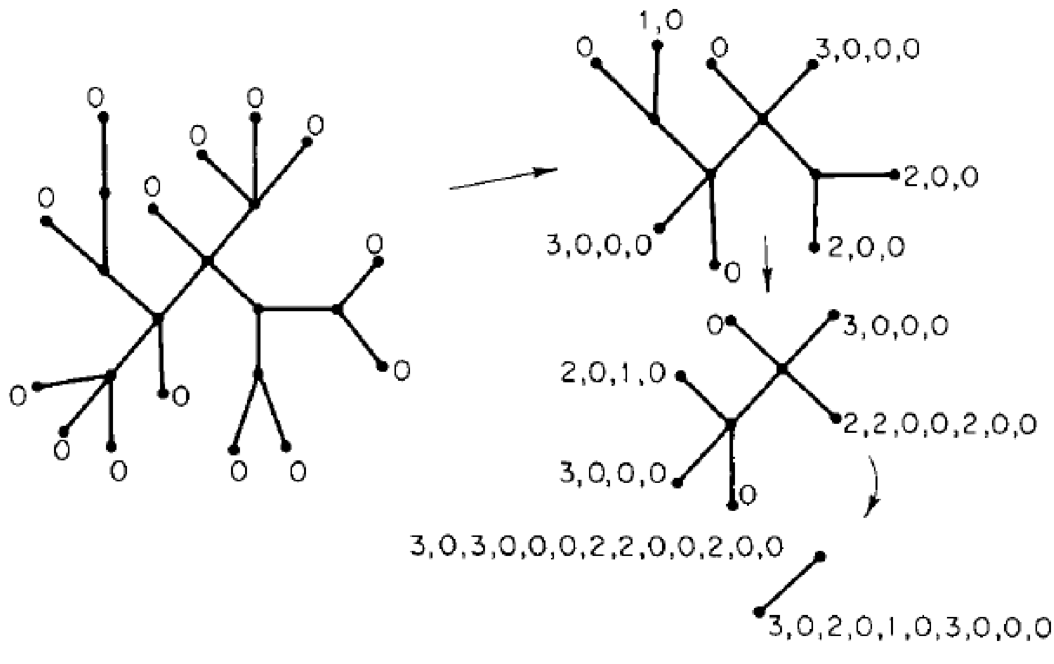


Рисунок 5 – Пример построения WAV-кода

Теорема 3. Если последовательность $\{u_i\}$ является WAV-кодом дерева, то для каждого r $(= 1, 2, \dots, p)$ ($p -$ количество вершин в дереве)

$$\sum_{i=1}^{r} u_i \geq r - 1, \tag{6}$$

где равенство выполняется тогда и только тогда, когда $r = p$.

Эта теорема используется для декодирования WAV-кодов.

Далее рассматривается код Прюфера. Алгоритм построения: выбирается лист с минимальным номером, записывается номер вершины смежной с минимальным листом, лист удаляется. Так происходит, пока не останется 2 вершины. Дерево, изображенное на рисунке 6, имеет такой код Прюфера: (7, 3, 7, 8, 2, 7, 5, 3, 8, 3, 12).

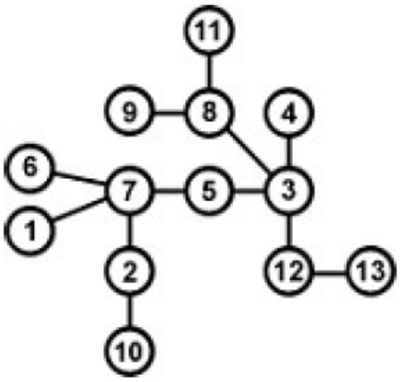
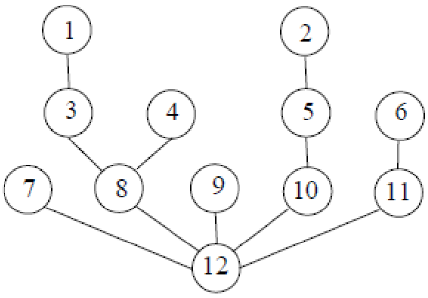


Рисунок 6 – Пример дерева для кода Прюфера

Канонический двоичный код – последовательность из 0 и 1 в определенном порядке. Он строится по следующим правилам. Если x – лист (но не корень), то $c(x) = \lambda$, где λ – пустое слово. Код вершины x, не являющейся листом, определяется после того, как построены коды всех ее сыновей. Пусть $\alpha_1, \alpha_2, \dots, \alpha_k$ – эти коды, упорядоченные лексикографически: $\alpha_1 \le \alpha_2 \le \dots \le \alpha_k$. Тогда $c(x) = 0\alpha_1 1 0\alpha_2 1 \dots 0\alpha_k 1$. Код корня и является двоичным кодом дерева. На рисунке 7 показаны дерево с корнем в вершине 12 и коды его вершин.



1	λ
2	λ
3	01
4	λ
5	01
6	λ
7	λ
8	010011
9	λ
10	0011
11	01
12	0101000111001100100111

Рисунок 7 – Построение двоичного кода

WS-код (*weight sequences*) – последовательность целых чисел, заданная $ws[root\ T] = [size[root\ T]]; ws[w_1]; \dots; ws[w_k]$, где вершины $w_1, \dots, w_k$ являются потомками корня дерева, расположенными в неубывающем лексикографическом порядке их канонических кодов, а $size[v]$ – количество вершин в поддереве дерева T с корнем в вершине v. Канонический WS-код дерева – канонический код корня дерева. На рисунке 8 показан пример построения канонического WS-кода.

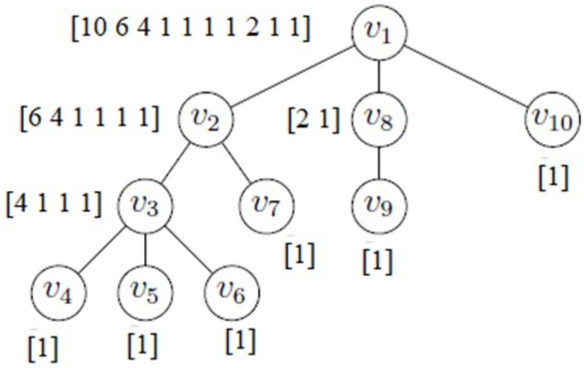


Рисунок 8 – Построение WS-кода

В шестом разделе рассматриваются алгоритмы генерации канонических уровневых и WS-кодов.

Алгоритм генерации канонических уровневых кодов состоит из двух процедур: «СЛЕД-ДЕРЕВО» и «СВОБОДНЫЕ-ДЕРЕВЬЯ». В процессе работы процедуры СВОБОДНЫЕ-ДЕРЕВЬЯ(n) осуществляется генерация всех главных канонических уровневых кодов n-вершинных свободных деревьев с использованием процедуры СЛЕД-ДЕРЕВО. Полный текст алгоритма в виде псевдокода приведен в Приложении А.

Алгоритм генерации канонических WS-кодов состоит из двух процедур: UFT(n) и BFT(n), которые используют вспомогательные процедуры RootedTrees(n) и RTHelper(n, q).

UFT(n) генерирует WS-коды для деревьев, размер поддеревьев корня которых не превышает $\frac{n-1}{2}$. BFT(n) генерирует остальные коды с размером поддеревьев корня равным $\frac{n}{2}$ (используется для четных n).

Функция RootedTrees(n) генерирует последовательность $B(n)$, используя вспомогательную функцию RTHelper(n, q), которая генерирует подпоследовательности $B_q(n)$, в которых вторым элементом является q.

Полный текст алгоритма в виде псевдокода приведен в Приложении Б.

В седьмом разделе рассматривается написанная программа, её интерфейс, возможности и показаны примеры работы.

Листинг программы приведен в Приложении В. Интерфейс программы показан на рисунках 9 и 10.

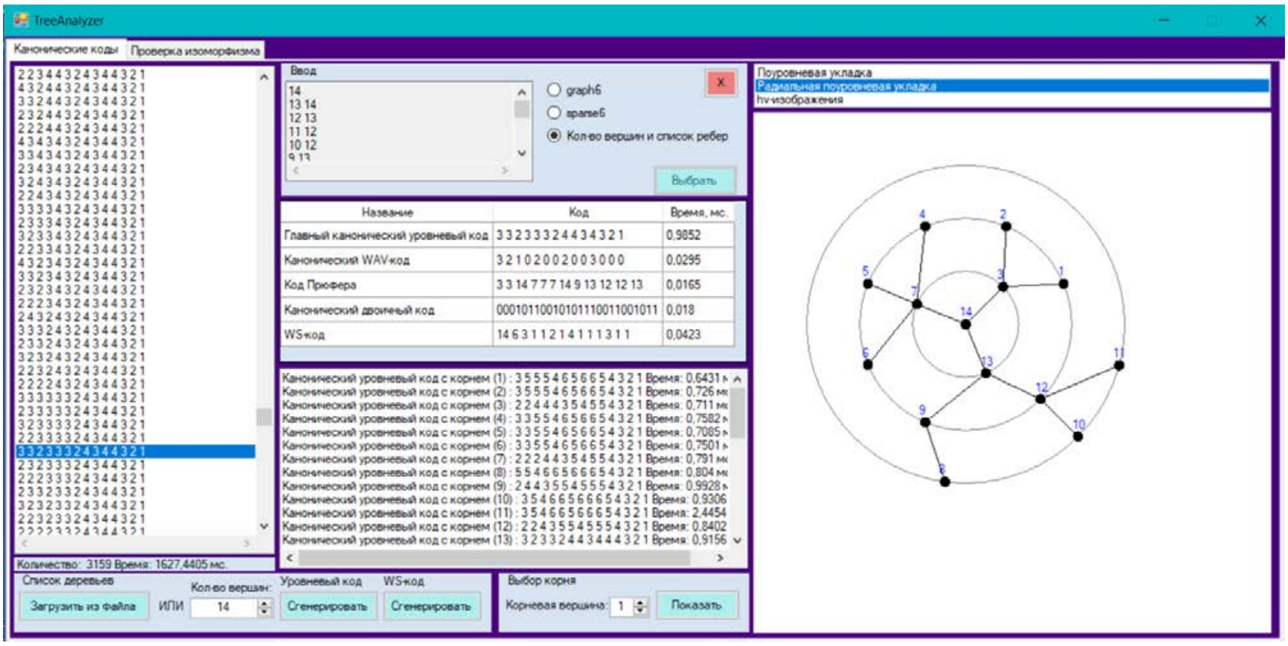


Рисунок 9 – Интерфейс программы. Канонические коды

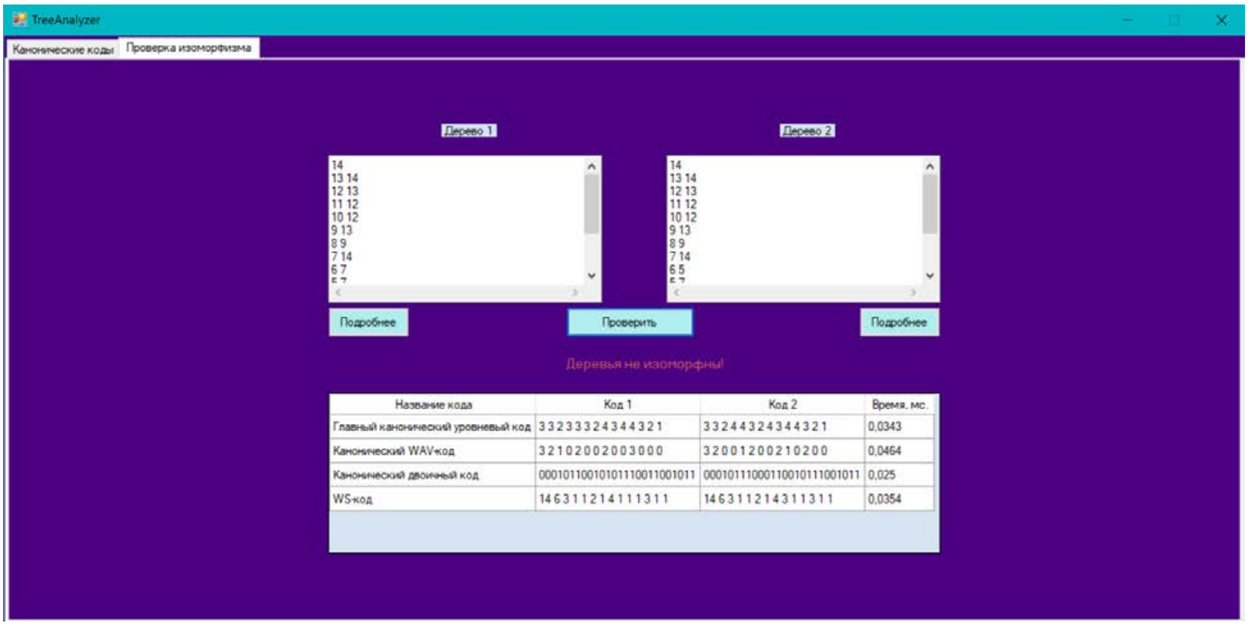


Рисунок 10 – Интерфейс программы. Проверка изоморфизма

В восьмом разделе приводятся результаты сравнения генерации уровневых канонических кодов и WS-кодов (таблица 1), число сгенерированных деревьев совпало с известным.

Таблица 1 – Проверка и сравнение времени генерации деревьев

n	Число деревьев	Время генерации уровневых кодов	Время генерации WS-кодов
1	1	0.2 мс.	0.6 мс.
2	1	0.2 мс.	0.7 мс.
3	1	0.2 мс.	0.7 мс.
4	2	0.5 мс.	0.9 мс.
5	3	0.6 мс.	1.2 мс.
6	6	1.4 мс.	1.8 мс.
7	11	2.5 мс.	2.9 мс.
8	23	5.3 мс.	5.2 мс.
9	47	17.7 мс.	19.8 мс.
10	106	48.9 мс.	51.4 мс.
11	235	113.5 мс.	118.7 мс.
12	551	275.8 мс.	277.4 мс.
13	1301	0.6 сек.	0.7 сек.
14	3159	1.5 сек.	1.6 сек.
15	7741	3.7 сек.	3.9 сек.
16	19320	9.2 сек.	9.2 сек.
17	48629	22 сек.	24 сек.
18	123867	57 сек.	1 мин.
19	317955	2.7 мин.	2.9 мин.
20	823065	9.5 мин.	10.5 мин.
21	2144505	49 мин.	51 мин.
22	5623756	3.5 ч.	4.3 ч.

ЗАКЛЮЧЕНИЕ

В современном мире большую роль играет информация, которая рассматривается как один из основных ресурсов информационного общества. Хранение, обработка и передача информации осуществляется с помощью информационных систем. Существует много различных способов описания и анализа информационных систем, в том числе использование теории графов.

В настоящее время есть необходимость компьютерного моделирования и проектирования сложных систем и изучения их свойств. Деревья являются хорошей математической моделью различных классов объектов и процессов.

В данной работе были рассмотрены канонические коды деревьев и генерация неизоморфных свободных деревьев с их помощью – канонические уровневые коды и WS-коды. Также в практической части в среде разработки Microsoft Visual Studio 2019 на языке C# на платформе .NET Framework 4.7.2 с использованием Windows Forms была написана программа для анализа деревьев, вычисления различных кодов деревьев и генерации неизоморфных свободных деревьев, дополнительная функция – проверка на изоморфизм двух деревьев. Программа реализует несколько способов отрисовки заданного дерева, с возможностью выбрать любую вершину в качестве корня дерева. Программа работает с несколькими форматами входных данных: graph6, sparse6, список ребер, загрузка списка в формате graph6 и sparse6 из файла или выбор из списка, сгенерированного с помощью канонических кодов.

Было проведено сравнение эффективности работы алгоритмов построения этих канонических кодов, которое показало, что уровневые канонические коды генерируются быстрее WS-кодов.

Таким образом, все поставленные задачи решены полностью.