

УДК 004.051:519.1

# СИСТЕМА КОНТРОЛЮ ПЛАГІАТУ В СТУДЕНТСЬКИХ РОБОТАХ

**В.І. Шинкаренко**

Доктор технічних наук, професор, доцент\*

Контактний тел.: 063-489-49-15

E-mail: Shinkarenko\_vi@ua.fm

**О.С. Куроп'ятник**

Інженер-програміст\*

Контактний тел.: 050-666-79-66

E-mail: elenadiit@rambler.ru

\*Кафедра «Комп'ютерні інформаційні технології»  
Дніпропетровський національний університет  
залізничного транспорту ім. акад. В. Лазаряна  
вул. ак. Лазаряна, 2, м. Дніпропетровськ, Україна,  
49010

*У статті наведено коротку характеристику програмних засобів для виявлення плагіату в текстах. Пропонується графовий підхід для моделювання та аналізу тексту на наявність плагіату. Наведено попередні результати розробки системи на основі вказаного підходу*

*Ключові слова: студентський плагіат, графова модель тексту*

*В статье приведена краткая характеристика программных средств для обнаружения плагиата в текстах. Предлагается графовый подход для моделирования и анализа текста на наличие плагиата. Приведены предварительные результаты разработки системы на основе указанного подхода*

*Ключевые слова: студенческий плагиат, графовая модель текста*

*In the article a brief description of software for detecting a plagiarism in texts is given. Approach on graphs for modeling and text analysis for the presence of plagiarism is proposed. The preliminary results of development of control system is presented, which base on the mentioned approach*

*Keywords: student plagiarism, graph model text*

## 1. Вступ

В умовах інтенсивного розвитку і реформування сфери освіти та науки все частіше постає питання якості вищої освіти, що безпосередньо пов'язане з проблемою плагіату в студентських роботах. З розвитком всесвітньої мережі Інтернет можливості плагіату значно розширилися. У вільному доступі з'явився великий обсяг науково-освітньої літератури, розробок студентів різних спеціальностей минулих років, та ресурсів, контент яких містить розв'язання задач, що можуть бути запозичені в цілому або фрагментами у лабораторних та курсових роботах.

Традиційно боротьба з плагіатом ведеться організаційними [1], правовими [2] та програмно-технічними [3] засобами.

Організаційні методи полягають у проведенні спецкурсів «Як уникнути плагіату», визначення відповідальних за контроль плагіату в студентських роботах. Для унеможливлення плагіату пропонується видача оригінальних завдань, щорічна їх зміна, введення рейтингової системи оцінювання для заохочення студентів до самостійної роботи [4].

Позитивні результати правових заходів можуть досягатися завдяки наявності відповідних договорів зі студентами [5].

Згідно проекту Закону України «Про вищу освіту» від 06.01.12, встановлення факту академічного плагіату є приводом для відрахування студента з вищого навчального закладу. Відповідно до проекту «вищі навчальні заклади зобов'язані вживати заходів (в тому числі шляхом запровадження відповідних новітніх технологій) щодо попередження та виявлен-

ня плагіату в наукових роботах ... здобувачів вищої освіти...».

Дана робота направлена на розвиток спеціалізованих програмно-технічних засобів.

## 2. Аналіз аналогів та постановка задачі

Наявність проблеми плагіату в різноманітних інформаційних ресурсах стало підставою розробки програмного забезпечення (ПЗ) для перевірки текстів на унікальність. Це такі програми та Інтернет сервіси, як AntiPlagiat.ru, пакет «Антиплагіат. ВУЗ», Etxt Антиплагіат, Advego Plagiat, Double Content Finder (DCFinder), Praide Unique Content Analyser 2, Miratools, Copyscape, istio.com та інші. Деякі можливості перевірки текстів закладено у найбільш поширеному текстовому редакторі MS Word.

Аналіз існуючого ПЗ показав, що у їх більшості не передбачається ведення додаткової, локальної бази оригінальних робіт (окрім «Антиплагіат. ВУЗ»), що не сприяє якісній перевірці студентських робіт на запозичення, які не публікувались в Інтернет - джерелах. У більшості аналогів відсутні наступні можливості:

- користування сервісом безпосередньо студентами;
- ведення обліку здачі робіт за кожною дисципліною;
- позначення переліку дисциплін, серед робіт яких необхідно провести пошук;
- пошук у споріднених дисциплінах;
- поповнення бази робіт при подачі останніх на перевірку;

– ведення обліку здачі робіт по кожному студентові із зазначенням всіх зданих робіт, строків здачі, відсотку плагіату;

– збереження статистичних даних [6].

Крім того робота з такими ПЗ передбачає необхідність зайвих дій викладача: він має вводити інформацію та у більшості випадків зберігати та аналізувати результати перевірки іншими засобами.

Однією з ключових проблем у вирішенні задачі контролю плагіату є великі обсяги інформації для порівняння та їх постійне зростання, що потребує наявності механізму регулювання поповнення та обмеження бази оригінальних робіт при перевірці.

Беручи до уваги вище сказане, можна зробити висновки, що більшість продуктів не орієнтовані на сферу освіти.

Таким чином задача даної роботи визначена як розробка методів пошуку та аналізу плагіату в студентських роботах, ефективних моделей представлення даних і проектування та розробка ПЗ на їх основі. ПЗ повинно мати таку функціональність: співставляти студентські роботи, мати механізм збору інформації для порівняння, максимально усунути викладача від перевірки робіт на плагіат, фіксувати факти плагіату, визначати його ступінь, частоту, виконувати моніторинг ситуації щодо плагіату в навчальному закладі.

### 3. Гафові моделі для представлення та порівняння текстів

Великі обсяги інформації для порівняння потребують ефективних алгоритмів та структур даних (СД). У зв'язку з цим пропонується моделювання тексту графовими засобами та їх представлення у ПЗ відповідними структурами.

Структура має відповідати таким вимогам:

- необхідно забезпечити взаємно однозначну відповідність між структурою даних і початковим текстом;
- час пошуку даних у структурі має бути якомога меншим;
- час формування структури має бути мінімальним;
- пам'ять, необхідна для зберігання структури, має бути мінімальною.

Вимоги зазначено у порядку їх важливості.

Розглянемо графову модель тексту.

Нехай вхідний текст – це впорядкований набір символів  $T = [t_i]$ ,  $i \in N_0$ , де  $t_i$  – і-ий друкований символ тексту, включаючи пробіл.

Представимо вхідний текст у вигляді множини орієнтованих навантажених графів [7], кожен з яких відповідає n-му унікальному символу ( $t_0$ ) тексту:

$$G_n(T) = \langle \{\bar{g}_i\}, \{\bar{d}_j\} \rangle, \quad \bar{g}_i = \langle g_i, t_i \rangle, \quad \bar{d}_j = \langle g_p, g_q \rangle, \\ i, j, p, q \in N_0;$$

де  $\{\bar{g}_i\}$  – множина навантажених вершин  $\bar{g}_i$  (ідентифікованих як  $g_i$ ),  $\{\bar{d}_j\}$  – множина дуг  $\bar{d}_j$ , початкова вершина  $g_0$  навантажена символом  $t_0$ :  $\bar{g}_0 = \langle g_0, t_0 \rangle$ ; кожному символу тексту  $t_i \neq t_0$  відповідає вершина  $\bar{g}_i = \langle g_i, t_i \rangle$  та дуга  $\bar{d}_i = \langle g_{i-1}, g_i \rangle$ . Якщо  $t_i = t_0$ , тоді  $\bar{g}_i = \bar{g}_0$  та відповідна дуга  $\bar{d}_i = \langle g_{i-1}, g_i \rangle = \langle g_{i-1}, g_0 \rangle$ .

Сформована таким чином структура в загальному вигляді представлена на рис. 1а. На рис. 1б показано приклад графу (приклад 1) для фрагменту тексту «шла саша по шоссе».

Для наведеної структури взаємно однозначна відповідність з початковим текстом не забезпечена. Початковим текстами для графу на рис. 1б можуть бути «ша пошла сашоссе», «шла саша саша по шоссе» та інші.

Для усунення цього недоліку введемо індекс шляху, що буде вказувати на порядок обходу графу. Оскільки вхідний текст може містити повторювані фрагменти, то відповідні вершини можуть повторюватися декілька разів, утворюючи ланцюжки, складові яких містять декілька індексів шляху. У загальному випадку навантажена дуга може бути представлена як  $\bar{d}_i = \langle g_i, g_{i+1}, W_i \rangle$ , де  $W_i = \langle w_{i,j_1}, w_{i,j_2} \dots w_{i,k(W_i)} \rangle$  – впорядкована множина індексів  $w_{i,j} \in N_0$ , до яких входить дуга,  $k(W_i)$  – кількість індексів в  $W_i$  (рис. 2а). На рис. 2а  $w_{0,j_1} = w_{1,j_1}$ ,  $w_{0,j_2} = w_{1,j_2}$ ,  $w_{0,j_3} = w_{n+1,j_1}$ ,  $w_{n,j_1} = w_{n+1,j_1}$ .

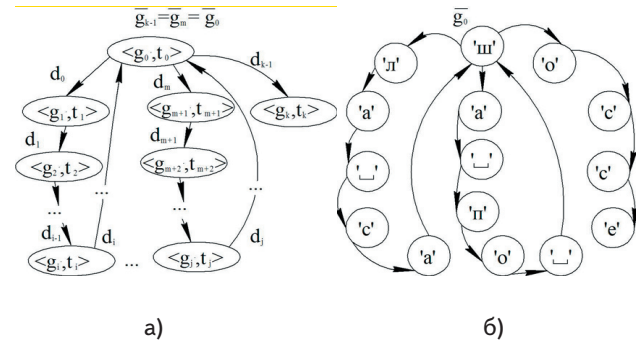


Рис. 1. Граф  $G_n(T)$  моделі тексту: а – загальної моделі; б – граф до приклада 1

Позначимо  $\tau(\bar{g}_p)$  – навантаження вершини  $\bar{g}_p$ ,  $f(T, t_0, i) - 1$  – кількість символів  $t_0$  у тексті  $T$  до  $i$ -ої позиції.

Модифікована графова модель тексту  $\{G'_n(T)\}$  складається з графів:

$$G'_n(T) = \langle \{\bar{g}_i\}, \{\bar{d}_j\} \rangle, \quad \bar{g}_i = \langle g_i, t_k \rangle, \quad \bar{d}_j = \langle g_p, g_q, W_j \rangle, \\ i, j, k, p, q \in N_0,$$

де початкова вершина  $g_0$  визначається як і для графу  $G_n(T)$ . Для кожного символу  $t_i$  такого, що  $t_{i+1} \neq t_0$  та  $\exists \bar{d}_j = \langle g_q, g_p, W_j \rangle : \tau(g_q) = t_i$  &  $\tau(g_p) = t_{i+1}$  має місце  $k(W_j) = k(W_r) + 1$ ,  $w_{j,k(W_j)} = f(T, t_0, i) - 1$ , де  $r : \exists \bar{d}_r = \langle g_s, g_p, W_r \rangle$ ,  $g_s$  – деяка вершина графу. Якщо  $t_{i+1} \neq t_0$  і  $\exists \bar{d}_j = \langle g_q, g_p, W_j \rangle : \tau(g_q) = t_i$  &  $\tau(g_p) = t_{i+1}$ , то існує  $\bar{g}_s = \langle g_s, t_i \rangle$  та дуга  $\bar{d}_u = \langle g_q, g_s, W_u \rangle$ ,  $k(W_u) = 1$ ,  $w_{u,1} = f(T, t_0, i) - 1$ .

У випадку  $t_{i+1} = t_0$ , то є дуга  $\bar{d}_u = \langle g_q, g_0, W_u \rangle$ ,  $k(W_u) = k(W_r) + 1$ ,  $w_{u,k(W_u)} = f(T, t_0, i) - 1$ .

Для наочності розглянемо графове представлення тексту «Мама мыла мылом раму» (приклад 2, рис. 2б). Текст містить повтор фрагменту «мыл», що можна відмітити за допомогою подвійного індексу шляху. Таким чином кожне розгалуження містить свій набір індексів шляху, що забезпечує однозначність.

Графова модель тексту  $\{G'_n(T)\}$  забезпечує перший критерій якості СД – взаємно однозначну відповідність між СД і початковим текстом. Для поліпшення другого критерію – часу пошуку в СД окремих послі-

довностей символів подальша модифікація моделі полягає у впорядкуванні (сортуванні) вершин графів.

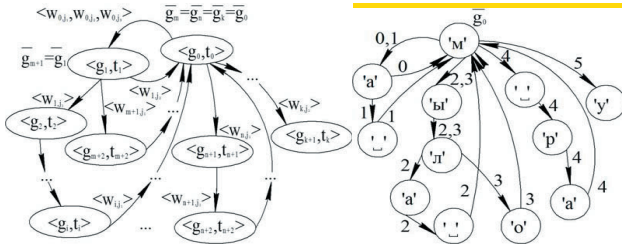


Рис. 2. Граф  $G'_n(T)$  моделі тексту: а – загальної моделі; б – граф до прикладу 2

У графовій моделі  $\{G''_n(T)\}$  існує взаємно однозначна відповідність вершин та дуг моделі  $\{G'_n(T)\}$ :

$$(\forall \bar{g}_i \in G'_n(T) \exists \bar{g}_j \in G''_n(T)) \& (\forall \bar{g}_i \in G''_n(T) \exists \bar{g}_j \in G'_n(T)) : \tau(\bar{g}_i) = \tau(\bar{g}_j)$$

та  $(\forall \bar{d}_i = \langle g_p, g_q, W_i \rangle \in G'_n(T) \exists \bar{d}_j = \langle g_k, g_l \rangle$

$$W_j \in G''_n(T)) \& (\forall \bar{d}_i \in G''_n(T) \exists \bar{d}_j \in G'_n(T))$$

:  $\tau(\bar{g}_p) = \tau(\bar{g}_k) \& \tau(\bar{g}_q) = \tau(\bar{g}_l) \& W_i = W_j$  та відсортовані нащадки кожної вершини за кодами символів  $\forall g_i, g_j : \exists (\bar{d}_p = \langle g_s, g_t, W_p \rangle \& \bar{d}_q = \langle g_s, g_t, W_q \rangle)$  при  $i < j$   $\tau(\bar{g}_i) < \tau(\bar{g}_j)$ .

У графовій моделі  $\{G'''_n(T)\}$  вершини навантажені рядками символів, тобто

$$G'''_n(T) = \langle \{\bar{g}'''_i\}, \{\bar{d}'''_j\} \rangle, \bar{g}'''_i = \langle g'''_i, E_k \rangle, \bar{d}'''_j = \langle g'''_p, g'''_q, W'''_j \rangle,$$

$i, j, k, p, q \in N_0$ , де  $E_k = [e_1, e_2, \dots, e_m]$  – масив (рядок) символів.

У графі  $G'''_n(T)$  вершини поділимо надві частини  $V''_1$  і  $V''_2$  такі, що  $g'''_k \in V''_1$  якщо

$$\exists \bar{d}'''_p = \langle g'''_k, g'''_q, W'''_p \rangle \& \bar{d}'''_q = \langle g'''_k, g'''_q, W'''_q \rangle : W'''_p = W'''_q$$

$V''_2 = V''' \setminus V''_1$ , де  $V'''$  – всі вершини графу. Визначимо взаємно однозначну відповідність між  $V''_1$  і  $V'''_1$  та між  $V''_2$  і  $V'''_2$  ( $V'''_1$  і  $V'''_2$  – частини графу  $G'''_n(T)$ ).

Взаємно однозначна відповідність вершин  $V''_2$  і  $V'''_2$  та відповідних дуг:

$$(\forall g'''_i \in V''_2 \exists g'''_j \in V'''_2 : \tau(g'''_i) = \tau(g'''_j)) \& (\forall g'''_i \in V''_2 \exists g'''_j \in V'''_2 : \tau(g'''_i) = \tau(g'''_j))$$

та суміжних дуг

$$(\forall \bar{d}'''_i = \langle g'''_p, g'''_q, W'''_i \rangle : g'''_p, g'''_q \in V''_2 \exists \bar{d}'''_j = \langle g'''_p, g'''_q, W'''_j \rangle : g'''_p, g'''_q \in V'''_2 \& \tau(g'''_p) = \tau(g'''_p)) \& (\forall \bar{d}'''_i = \langle g'''_p, g'''_q, W'''_i \rangle : g'''_p, g'''_q \in V''_2 \exists \bar{d}'''_j = \langle g'''_p, g'''_q, W'''_j \rangle : g'''_p, g'''_q \in V'''_2 \& \tau(g'''_p) = \tau(g'''_p))$$

У  $V'''_1$  поєднані вершини  $V''_1$ :

$$\text{Для } \forall g'''_i, g'''_j, \dots, g'''_m \in V''_1 : \bar{d}'''_i = \langle g'''_i, g'''_j, W'''_i \rangle, \bar{d}'''_j = \langle g'''_i, g'''_j, W'''_j \rangle, \dots, \bar{d}'''_{m-1} = \langle g'''_i, g'''_j, W'''_{m-1} \rangle \& W'''_i = W'''_j = \dots = W'''_m$$

$$\exists \bar{g}'''_r = \langle g'''_i, E_i \rangle,$$

$$E_r = \tau(\bar{g}'''_i) + \tau(\bar{g}'''_j) + \dots + \tau(\bar{g}'''_m),$$

де «+» – операція конкатенації.

$$\text{Для } \forall \bar{d}'''_r = \langle g'''_i, g'''_j, W'''_r \rangle,$$

$$\exists \bar{d}'''_r = \langle g_s, g_t, W'''_r \rangle, \& W'''_r = W'''_r$$

$$\forall \bar{d}'''_r = \langle g_r, g_s, W'''_r \rangle \exists \bar{d}'''_s = \langle g_r, g_s, W'''_s \rangle, W'''_r = W'''_s$$

$$\text{Зворотно } \forall \bar{g}'''_i = \langle g'''_i, E_i \rangle \in V'''_1$$

$$\exists \bar{g}'''_1, \bar{g}'''_2, \dots, \bar{g}'''_m \in V'''_1 : \bar{g}'''_1 = \langle g_{j_1}, t_{j_1} \rangle,$$

$$\bar{g}'''_2 = \langle g_{j_2}, t_{j_2} \rangle, \dots, \bar{g}'''_m = \langle g_{j_m}, t_{j_m} \rangle,$$

$$E_i = \tau(\bar{g}'''_1) + \tau(\bar{g}'''_2) + \dots + \tau(\bar{g}'''_m) \&$$

$$\exists \bar{d}'''_1, \bar{d}'''_2, \dots, \bar{d}'''_m : \bar{d}'''_1 = \langle \bar{g}'''_1, \bar{g}'''_2, W'''_1 \rangle, \bar{d}'''_2 =$$

$$= \langle \bar{g}'''_2, \bar{g}'''_3, W'''_2 \rangle, \bar{d}'''_m = \langle \bar{g}'''_m, \bar{g}'''_1, W'''_m \rangle$$

$$\& W'''_1 = W'''_2 = \dots = W'''_m \text{ та для } \forall \bar{d}'''_r = \langle \bar{g}'''_r, \bar{g}'''_r, W'''_r \rangle$$

$$\exists \bar{d}'''_r = \langle \bar{g}'''_r, \bar{g}'''_r, W'''_r \rangle : W'''_r = W'''_r \text{ і } \forall \bar{d}'''_r = \langle \bar{g}'''_r, \bar{g}'''_r, W'''_r \rangle$$

$$\exists \bar{d}'''_r = \langle \bar{g}'''_r, \bar{g}'''_r, W'''_r \rangle : W'''_r = W'''_r$$

На рис. За приведений граф  $G'''_n(T)$  у загальному вигляді та для прикладу 2.

Модифікація  $\{G'''_n(T)\}$  до виду  $\{G''''_n(T)\}$  була виконана для забезпечення ще одного критерію СД – пам'ять, необхідна для зберігання структури, має бути мінімальною.

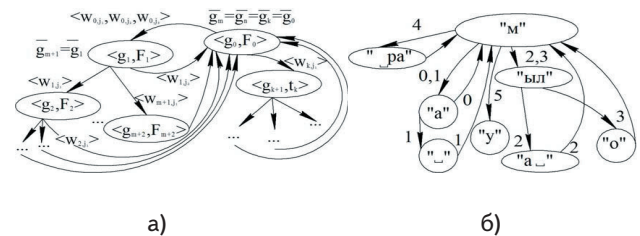


Рис. 3. Граф  $G''''_n(T)$  моделі тексту: а – загальної моделі; б – стиснутий граф до прикладу 2

Таким чином запропонована графова модель  $\{G''''_n(T)\}$  дозволяє однозначно представляти структуру тексту, а також впорядковувати підрядки, що містяться в тексті таким чином, що для кожного символу або послідовності можна визначити всі можливі доповнення, що містяться у вхідному тексті. Це в свою чергу, значно полегшує співставлення вхідного тексту поданого на перевірку у вигляді рядка, та оригінального тексту, представленого як набір графів, які мають різні початкові вершини.

Визначимо обчислювальну складність алгоритму порівняння з використанням запропонованої моделі. Нехай маємо вхідний ланцюжок  $l(n)$ , де  $n$  – довжина ланцюжка. Маємо множину графів  $\{G'_i(T)\}$ ,  $i \in N_0$ . Граф будується для кожного унікального друкованого символу та пробілу з тексту роботи-зразка. Нехай дана множина впорядкована за символьним навантаженням стартових вершин  $\bar{g}'''_0$ . Перехід від  $j$ -ої вершини до  $(j+1)$ -ої відбувається по першій знайденій дузі. Виходячи з умови, що навантаження вершини співпадає з відповідним символом ланцюжка  $l$  і  $l \in G$ , даний випадок можна вважати кращим. Тоді алгоритм має лінійну складність  $O(n)$ . За умови, що першим обраний граф для порівняння має навантаження по-

чаткової вершини рівне першому символу ланцюжка, то складність вибору графу дорівнює 1. Тоді загальна складність алгоритму:  $O(n) = n$ .

Найгіршим випадком є ситуація, коли граф містить ланцюжки, довжина яких не перевищує  $n/2$  і зміщенні відносно першого символу 1 і один одного на одиницю. Тоді кількість операцій для співставлення ланцюжка буде рівною  $\frac{n}{2} \cdot \frac{n}{2}$ , де перший множник – довжина ланцюжка, що співпав, другий – кількість ланцюжків. Складність операції пошуку необхідного графу для порівняння при використанні бінарного пошуку можна оцінити як  $\log_2(n)$ . Тоді у найгіршому випадку загальна складність алгоритму пошуку ланцюжка в графі визначається як  $O(n) = n^2$ .

На даний момент виділено три типи користувачів: адміністратор, викладач, студент. На адміністратора покладена задача реєстрації користувачів у системі.

Для студента передбачено такі функції, як подання роботи на перевірку та перегляд результатів задачі всіх робіт. Для кожної функції передбачено відповідний інтерфейс (рис. 4).

Для викладача передбачено такі функції, як перегляд результатів задачі всіх робіт за вказаний період та по заданій дисципліні та встановлення поміток про задачу роботи. Для кожної функції передбачено відповідний інтерфейс (рис. 5).

Результати порівняння робіт представляються у виді «Кількість спроб задачі/найменший відсоток плагіату/відмітка про прийняття системою».

Дисципліна	Тип	№	Кількість спроб задачі	Кращий рез-т (% плагіату)	Прийнято	Здано
ПЗ локальних мереж	лабораторна	1	1	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ПЗ локальних мереж	лабораторна	2	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Розробка ЕС та їх ПЗ	лабораторна	2	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Розробка ЕС та їх ПЗ	лабораторна	1	1	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Розробка ЕС та їх ПЗ	лабораторна	3	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Сист. ПЗ сучасних систем	лабораторна	1	1	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 4. Вікно користувача-студента: представлення загальних результатів перевірки

Для визначення часової ефективності алгоритмів [8] порівняння текстових файлів виконано комп'ютерний експеримент. Вимірювався час порівняння 150 випадково відібраних текстових файлів кожен з кожним (розміром 14... 1185 Кб, з 1...23 сторінок) розробленим ПЗ та MS Word 2007. Експеримент виконано на сервері з такими характеристиками: процесор Intel(R) Xeon(R) CPU 3065, кеш L1 кода/ L1 даних/ L2 – 32/32/4096 Кб, тактова частота/частота системної шини/частота пам'яті – 2,33 ГГц (3,5\*667)/1333 МГц/400 МГц, час доступу до ОП (читання/запис) 6527/2632 Мб/с, операційна система – MS Windows XP Prof SP3. Розрахункові показники оцінки ступеня переваги склали 97,8%, області переваги – 99,5% визначили подавляючи перевагу розробленого авторами алгоритму обробки СД.

Сдані роботи  
Розробка ЕС та їх ПЗ

П.І.Б.	Пр №1	Пр №2	Пр №3	Пр №4	Пр №5	Курсова
Куроп'ятник О. С.	1/0/ прийнято	1/0/ прийнято	1/0/ прийнято			
Петров П. П.	2/74/ неприйнято					
Іванов І. І.	1/96/ неприйнято					

ПЗ локальних мереж

П.І.Б.	Пр №1	Пр №2	Курсова
Куроп'ятник О. С.	1/0/ прийнято	1/0/ прийнято	

Рис. 5. Представлення результатів перевірки робіт у профілі викладача

На сьогодні система потребує тестування блоку співставлення з метою перевірки правильності роботи та визначення її часових характеристик в умовах навчального процесу. Триває робота з удосконалення та нарощення спеціальної бази даних, що міститиме в собі тексти у графовому представленні та відомості щодо авторів текстів та належності до певної категорії відповідно до дисциплін, що дозволить задавати категорію пошуку та при виявленні факту наявності плагіату визначати його джерело.

Розширення бази інформацією про студентів дозволить вести облік робіт з вказівками на кількість, частоту та обсяги плагіату у роботах студентів. Також передбачається поповнення бази текстів останніми екземплярами робіт, зданими користувачами на перевірку, що забезпечує постійне поповнення матеріалів для порівняння.

#### 4. Попередні результати розробки програмного забезпечення на основі обраного підходу

Для апробації графової моделі та співставлення текстів на її основі було розроблено програму для перевірки студентських робіт на автентичність. Дана програма охоплює такі функціональні аспекти:

- виявлення запозичених фрагментів у студентських роботах;
- визначення обсягів плагіату, його відсотку від об'єму роботи, середнього відсотку плагіату в роботах студента;
- ведення та поповнення бази оригінальних робіт за рахунок тих, які були подані на перевірку.

#### Висновки

Розробка графової моделі тексту дозволила у компактному вигляді представити вхідні тексти оригінальних робіт, згрупувавши символи у підрядки з однаковими префіксами з метою прискорення їх аналізу.



Створення локальної бази робіт у межах кафедри (і університету в майбутньому) дозволило значно зменшити обсяги інформації.

Так як частина студентських робіт є запозиченою з Інтернет-ресурсів, то включення таких робіт до локальної бази дозволить значною мірою охопити зовнішні інформаційні ресурси, уникнувши багаторазового повторення їх контенту.

Організація системи з декількома типами користувачів дозволяє усунути викладача від виконання функції контролю плагіату. ПЗ надає студенту зручний спосіб аналізу робіт на унікальність з метою підвищення якості власних робіт з точки зору сторонніх запозичень та доведення факту автентичності роботи. Розроблене ПЗ має сприяти вирішенню проблеми плагіату, стимулюючи самостійну роботу студентів над завданнями.

Література

1. Голунов, С.В. Студенческий плагиат как вызов системе высшего образования в России и за рубежом [Текст] / С. В. Голунов // Вопросы образования. – 2010. – № 3. – С. 243-257.
2. Савочкина, Т. С. Студенческий плагиат в высшем профессиональном образовании / Т. С. Савочкина, И. В. Нератова // Вестник Тверского государственного университета. Серия: Педагогика и психология. – 2009. – № 6. – с. 90–104.
3. Kakkonen, T., Myller, N. A Sampling-based Tool for Plagiarism Detection in Student Texts // Proceedings of the 8th European Conference on e-Learning. – Bari, Italy. – 2009.
4. Болкунов, И. А. Пути преодоления студенческого плагиата [Электрон.ресурс]. – Режим доступа: <http://www.nbu.gov.ua>.
5. Иванова, А. А. Проблема студенческого плагиата [Текст] / А. А. Иванова // Преступление против интеллектуальной собственности: Материалы Международной научно-практической конференции (Нижний Новгород, 19-20 мая 2011 года), г. Нижний Новгород: НИУ ВШЭ - Нижний Новгород, 2011.
6. Мокін, В. Б. Автоматизована система перевірки текстів на плагіат [Текст] / В. Б. Мокін, С. В. Бевз, О. В. Говенко, І. А. Білоус // Вісник Вінницького політехнічного інституту. – 2010. – № 5. – с. 12 – 17.
7. Кристофидес, Н. Теория графов [Текст] / Н. Кристофидес. – М.: Мир. – 1978. – 432с.
8. Шинкаренко, В. И. Экспериментальные исследования алгоритмов в программно-аппаратных средах : монография / В. И. Шинкаренко. – Д.: Изд-во Днепропетр. нац. ун-та ж.-д. трансп. им. акад. В. Лазаряна, 2009. – 279 с.

**Одне з перших правил Е-комерції: якщо користувач не знайде товар, то він його не купить. Тому проектування швидких і зручних користувачеві інтерфейсів веб-додатків є життєвою необхідністю**

**Ключові слова: юзабиліти, людино-машинна взаємодія, евристика**

---

**Одно из первых правил Е-коммерции: если пользователь не найдет товар, то он его не купит. Поэтому проектирование быстрых и удобных пользователю интерфейсов веб-приложений является жизненной необходимостью**

**Ключевые слова: юзабилити, человеко-машинное взаимодействие, эвристика**

---

**One of the first rules of E-commerce: if users cannot find the product, they will not buy it. Therefore, fast and human-centered interfaces design of web-applications is a necessity**

**Keywords: Human-Computer Interaction, Usability, Heuristics**

УДК 004.5

# ПРИМЕНЕНИЕ ПРИНЦИПОВ ЧЕЛОВЕКО-ОРИЕНТИРОВАННОГО ИНТЕРФЕЙСА В WEB-ПРИЛОЖЕНИЯХ

**Р. В. Мельникова**  
 Кандидат технических наук, доцент  
 Кафедра программной инженерии  
 Харьковский национальный университет  
 радиоэлектроники  
 пр. Ленина, 14, г. Харьков, Украина, 61166  
 Контактный тел.: (057) 702-14-46, (057) 702-13-50  
 E-mail: melnikova@kture.kharkov.ua

## 1. Постановка проблемы

Пользователи в сети очень целенаправленны. Если веб-сайт трудно использовать, люди уходят. Если пользователи теряются на сайте и не могут

получить ответы на ключевые вопросы, они уходят. При наличии других сайтов, предоставляющих аналогичные услуги, следование принципам и правилам веб-юзабилити становится жизненно необходимым.