1. Курсова робота на тему «Перевірка на плагіат програмного коду лабораторних робіт студентів». Виконав студент групи МІ-3 Шевченко Максим. Науковий керівник – Панченко Тарас Володимирович.
2. Проблема використання чужої праці досить поширена серед студентів. На фото наведено приклад трьох ідентичних лабораторних робіт від різних студентів, які знайшов викладач.
3. **Актуальність** роботи пояснюється поширеним використанням плагіату серед студентів та відсутністю системи обліку лабораторних робіт в університеті.

**Метою** роботи є реалізація алгоритмів порівняння програмних кодів на плагіат із можливістю їх подальшої інтеграції в систему обліку лабораторних робіт.

**Об’єктом** роботи є організація перевірки лабораторних робіт в навчальному закладі і підходи до розробки програмного забезпечення для цих цілей.

**Предметом** роботи є дослідження способів маскування плагіату та алгоритмів його виявлення.

1. Хоча існують різні системи перевірки плагіату як у текстових роботах, так і в програмному коді, проте загальнодоступних спеціалізованих саме на лабораторних роботах систем немає. Інакше вони б використовувалися в усіх навчальних закладах.
2. У своїй роботі я вирішив застосувати алгоритми пошуку плагіату з використанням токенізації коду, тобто його представленні послідовністю лексем. Такі алгоритми добре визначають плагіат та є простими в реалізації.

При застосуванні токенізованих алгоритмів спочатку необхідно виконати лексичний аналіз і звести код до певного стандартного вигляду, тобто виконати його нормалізацію. Після цього до отриманого нормалізованого коду застосовують методи порівняння рядків, як от пошук найдовшого спільного підрядка та метод шинглів.

1. Я розробив власний алгоритм нормалізації коду, який, на мою думку, є оптимальним для пошуку плагіату в лабораторних роботах. Розглянемо його на прикладі простої програми додавання двох чисел.
2. Спочатку видалимо всі коментарі та не програмний текст. Після цього, орієнтуючись по роздільникам мови, знайдемо використані слова й числа.
3. Виконаємо заміну ключових слів мови програмування на певні символи. Конкретно що на що треба заміняти має бути вказано у зовнішньому файлі. Після цього виконаємо заміну чисел на токен числового типу, наприклад int.
4. Решті слів надаємо значення найближчого до них токена та видаляємо пробіли й абзаци. Після цих операцій до нормалізованого коду застосовують алгоритми порівняння текстів.
5. Найдовшим спільним підрядком називають підрядок найдовшої довжини, який міститься в усіх рядках, які порівнюємо. Для його пошуку серед двох рядків застосовують описаний на слайді алгоритм заповнення матриці. При його використанні ми знаходимо максимальні довжини підрядків, які починаються з кожного символу s1.
6. У моїй реалізації я виконую пошук і видалення найбільшого спільного підрядка до тих пір, поки не залишиться лише оригінальна частина коду. Коефіцієнт плагіату знаходиться через відношення довжини оригінальної частини до всієї довжини коду.
7. **Алгоритм шинглів** розроблений для пошуку копій та дублікатів тексту у веб-документі. Є засобом для виявлення плагіату.

**Шингли** – це виділені з тексту підпослідовності слів.

1. Для забезпечення кращої роботи алгоритму із лабораторними роботами було вирішено використовувати весь набір шинглів, а не їх частину. Також коефіцієнт плагіату визначається як відношення розміру перетину наборів шинглів обох кодів до розміру набору шинглів програми, підозрілої на плагіат.
2. Наступний слайд містить опис результатів виконання реалізованих алгоритмів. Замість нього можна переглянути відео з демонстрацією роботи програми, де результати теж вказані. Проте перегляд відео потребує додаткового часу до виділеного на доповідь (+1хв).
3. Розглянемо результат реалізації алгоритмів на практиці. Тут s1 і s2 – стара й нова версії нормалізатора, s3 – лабораторна з криптографії. Як можна помітити при порівнянні однакових робіт коефіцієнт плагіату рівний одиниці, при перевірці подібних робіт він знижується, проте все одно є досить значним, а при порівнянні різних робіт прямує до нуля. Це засвідчує коректність роботи реалізованої системи.
4. У результаті виконання роботи було досліджено способи маскування плагіату й алгоритми його виявлення. А також було реалізовано власний алгоритм токенізації та нормалізації коду, алгоритми шинглів та пошуку найдовшого спільного підрядка, модифіковані під поставлене завдання. Завдяки модульній структурі програми (окремі файл з граматикою мови, класи нормалізатора та перевірки плагіату) можлива зміна та покращення її окремих частин без шкоди іншим компонентам. До того ж, реалізовані алгоритми можуть застосовуватися не тільки для пошуку плагіату в коді програм, а й у звичайних текстових роботах. Отриманий продукт виконує поставлені на нього завдання та може бути інтегрований в систему обліку лабораторних робіт.
5. На цьому доповідь завершено. Дякую за увагу!