

Інформаційні технології

УДК 004.421.6

Войденко Олексій Костянтинович

студент

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

Войденко Алексей Константинович

студент

Национального технического университета Украины

«Киевский политехнический институт имени Игоря Сикорского»

Voidenko Oleksii

Student of the

National Technical University of Ukraine

«Igor Sikorsky Kyiv Polytechnic Institute»

Науковий керівник:

Кисляк Сергій Володимирович

старший викладач кафедри біомедичної кібернетики

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**ОПТИМІЗАЦІЯ АЛГОРИТМУ ЧЖУ-ТАКАОКИ ДЛЯ ЗАДАЧ
БІОІНФОРМАТИКИ**

**ОПТИМИЗАЦИЯ АЛГОРИТМА ЧЖУ-ТАКАОКИ ДЛЯ ЗАДАЧ
БИОИНФОРМАТИКИ**

**OPTIMIZATION OF ZHU-TAKAOKA ALGORITHM FOR
BIOINFORMATICS OBJECTIVES**

Анотація. В роботі розглянуті алгоритми пошуку підрядка в рядку та їх використання для задач біоінформатики. Виконаний аналіз ефективності алгоритмів для задач точного пошуку фрагментів біологічних послідовностей. Розглянуті алгоритм Бойера-Мура, алгоритм Райта, алгоритм Чжу-Такаокі та його оптимізація.

Ключові слова: оптимізація, алгоритми пошуку підрядка, біоінформатика, алгоритм Бойера-Мура, порівняння рядків.

Аннотация. В работе рассмотрены алгоритмы поиска подстроки в строке и их использование в биоинформатике. Выполнен анализ эффективности алгоритмов для задач поиска фрагментов биологических последовательностей. Рассмотрены алгоритм Бойера-Мура, алгоритм Райта, алгоритм Чжу-Такаока и его оптимизация.

Ключевые слова: оптимизация, алгоритмы поиска подстроки, биоинформатика, алгоритм Бойера-Мура, сравнение строк.

Summary. String-searching algorithms and their use in bioinformatics are considered. Completed analysis of the algorithms effectiveness for the objectives of exact searching for fragments of biological sequences. The Boyer-Moore algorithm, the Raita algorithm, the Zhu-Takaoka algorithm and its optimization are considered.

Key words: optimization, string-searching algorithms, bioinformatics, Boyer-Moore algorithm, string comparison.

Постановка проблеми. Алгоритми пошуку підрядка в рядку знайшли безліч застосувань. Наприклад, вони можуть бути застосовані для звичайного пошуку слова у словнику або ж пошуку фрагмента не описаної біологічної послідовності в референсному геномі з наступною можливістю перенесення опису. При застосуванні алгоритмів точного пошуку для задач біоінформатики необхідно враховувати деякі важливі особливості

щодо організації біологічних полімерів (амінокислотних та нуклеотидних послідовностей). Біологічні макромолекули записані з використанням маленького алфавіту, що складається з чотирьох нуклеотидів (символів) та формує довільний зразок біологічної послідовності. Референсна послідовність, як правило, має велику довжину, яка може складати мільярди нуклеотидів. Наприклад, геном людини записаний 3,2 мільярдами нуклеотидів. Враховуючи особливості біологічних послідовностей, для задач пошуку підрядка в рядку необхідно використовувати ефективні алгоритми [1].

Аналіз останніх досліджень і публікацій. Частково питання оптимізації пошуку підрядка в рядку розглядалися в працях Айткулова П.Г., Желудкова А.В., Солдатової Г.П., Сухачева А.И та інших.

Мета роботи. Розробити програмний застосунок, що реалізує оптимізований алгоритм Чжу-Такаокі для вирішення задачі пошуку фрагментів біологічних послідовностей в референсному геномі

Виклад основного матеріалу. Існує велика кількість алгоритмів, що дозволяють вирішити задачу пошуку підрядка в рядку, найпростіший з них – алгоритм “грубої сили”. Суть такого підходу полягає у покроковій підстановці вибраного зразка на всі можливі позиції зліва направо та порівнянні його з кожним сегментом послідовних символів у рядку, в якому виконується пошук [2]. Часова складність алгоритму $O((n-m+1)*m)$, де n – довжина рядка, а m – довжина зразка. Такий алгоритм не може бути застосований на текстах великої довжини.

Одним з перших алгоритмів, які ефективно використовували ресурси пам'яті та часу при вирішенні задачі пошуку підрядка в рядку, був алгоритм Бойера-Мура. Його особливостями є дві евристики та зіставлення рядків з кінця вікна порівняння [3].

1. Евристика стоп-символу.

На стадії препроцесингу створюється таблиця стоп-символів для усього алфавіту, який використовується у зразку. В ній позначена остання позиція кожного з можливих символів алфавіту в підрядку. Позначимо через x – шуканий зразок (підрядок), y – текст (рядок), у якому виконується пошук. Якщо стається невідповідність між символом $y[i+j]=b$ в тексті y і символом $x[i]=a$ підрядка x під час його підстановки на позицію j , то за допомогою евристики стоп-символу виконується зсув зразка вправо до тих пір, поки не відбудеться збіг між порівнюваними символами (рис. 1).

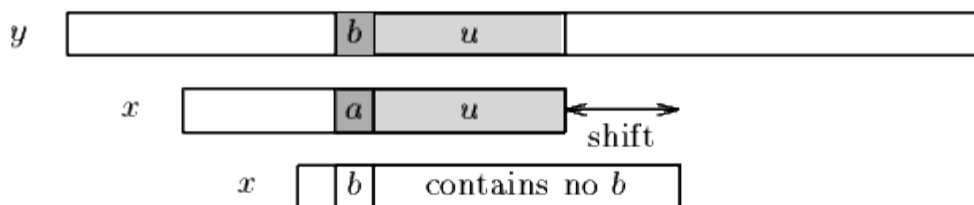


Рис. 1. Зсув за евристикою стоп-символу

Джерело: відповідно до [4]

2. Евристика збігу суфіксів.

На стадії препроцесингу створюється таблиця суфіксів для підрядка [5]. Для всіх суфіксів, які зустрічаються у зразку вказується найменше значення відстані, на яку потрібно зсунути підрядок вправо, щоб його найперший суфікс, який дорівнює v почав співпадати з сегментом символів тексту y , який також дорівнює v , зазначається, що для запобігання повторної невідповідності, символ перед v після зсуву не повинен дорівнювати символу, який стоїть перед v до зсуву (рис. 2).

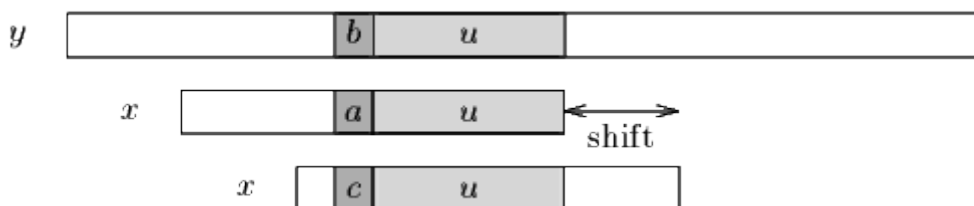


Рис. 2. Зсув за евристикою збігу суфіксів

Джерело: відповідно до [4]

Часова складність алгоритму Бойера-Мура в найкращому випадку $O(n/m)$. Деякі оптимізації Алгоритму Бойера-Мура гарантують часову складність $O(n)$ [6]. Однією з оптимізацій, що заслуговують увагу є алгоритм Райта.

Алгоритм Райта при кожній спробі спочатку порівнює останній символ підрядка з правим текстовим символом вікна порівняння, тоді, якщо вони збігаються, він порівнює перший символ зразка з лівим текстовим символом вікна порівняння, тоді якщо вони співпадають він порівнює середній символ підрядка з середнім символом у вікні порівняння. Якщо вони збігаються, то після цього порівнюють інші символи від другого до останнього, іноді порівнюючи середній символ знову. Райт зберіг евристику стоп-символу, але відмовився від евристики збігу суфіксів, тому що на практиці вона суперечить особливості роботи його алгоритму [7]. Цей метод є максимально ефективним при пошуку підрядків в англійській текстах.

Ще однією оптимізацією алгоритму Бойера-Мура є алгоритм Чжу-Такаокі. Його автори помітили, що евристика стоп-символу працює не належним чином при її використанні на малих алфавітах та змінили її так, щоб таблиця будувалася не для одного, а для двох символів. Цей алгоритм враховує евристику збігу суфіксу.

Якщо використовувати ці методи в такому вигляді, в якому їх представили автори, то виконати пошук фрагменту біологічної

послідовності, наприклад, в геномі людини, можна, але ефективність роботи алгоритмів буде погіршена відсутністю оптимізації. Якщо додати до ефективного алгоритму ключову особливість іншого схожого і також ефективного алгоритму, то теоретично можна досягнути ситуативних покращень в роботі нового оптимізованого алгоритму. Таким чином, було вирішено додати евристику порівняння останнього та першого символу (особливість алгоритму Райта) в алгоритм Чжу-Такаокі, але без порівняння середнього символу. Для подальшої оптимізації також була видалена евристика збігу суфіксу.

Було створено програмний додаток з реалізацією вище описаних алгоритмів та їх особливостей (рис. 3).

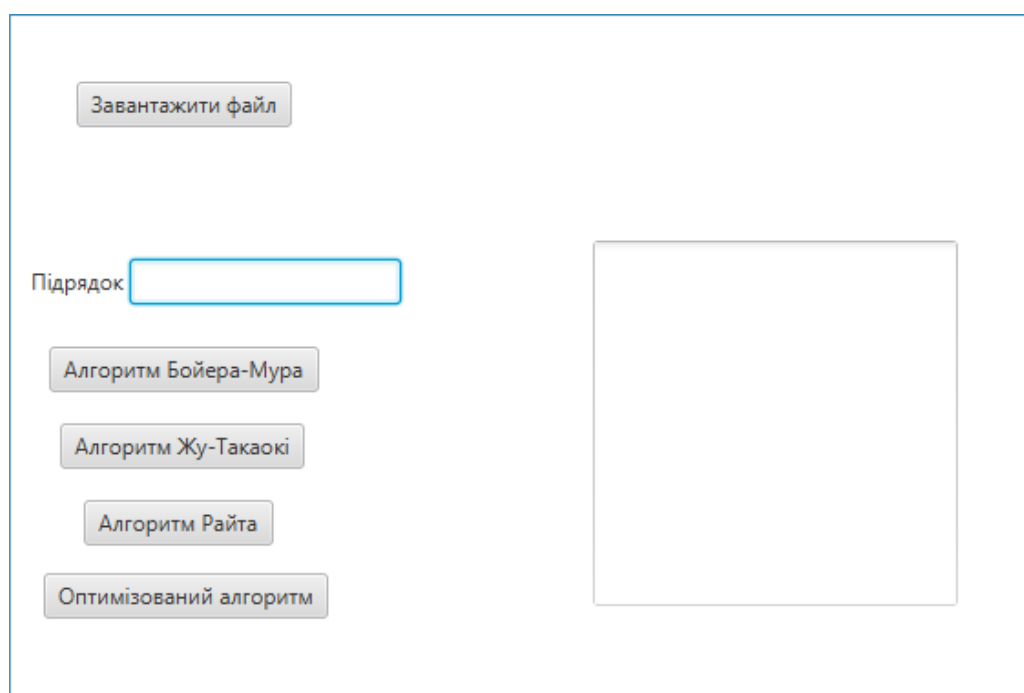
The image shows a web-based application interface. At the top left is a button labeled "Завантажити файл" (Upload file). Below it is a label "Підрядок" (Line) followed by a text input field. To the right of the input field is a large, empty rectangular box. Below the input field is a vertical list of five buttons: "Алгоритм Бойера-Мура" (Boyer-Moore algorithm), "Алгоритм Жу-Такаокі" (Zhu-Takao algorithm), "Алгоритм Райта" (Wright algorithm), and "Оптимізований алгоритм" (Optimized algorithm).

Рис. 3. Інтерфейс програми

Для початку роботи необхідно завантажити файл у FASTA форматі, який є одним із основних форматів збереження даних біологічних послідовностей. Після цього необхідно ввести підрядок для пошуку та вибрати один з алгоритмів для використання. В результаті роботи алгоритму можуть бути отримані індекси позицій, починаючи з

яких, символи підрядка повністю співпадають з символами рядка. Для проведення аналізу також були додані показники часу виконання алгоритму (рис. 4).

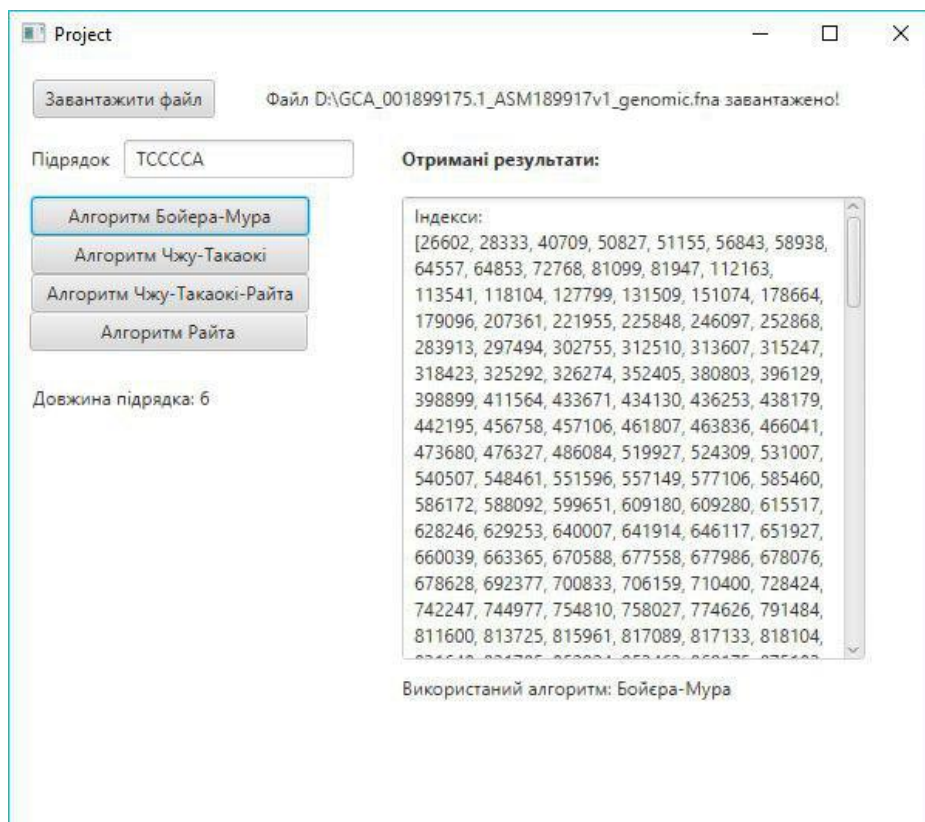


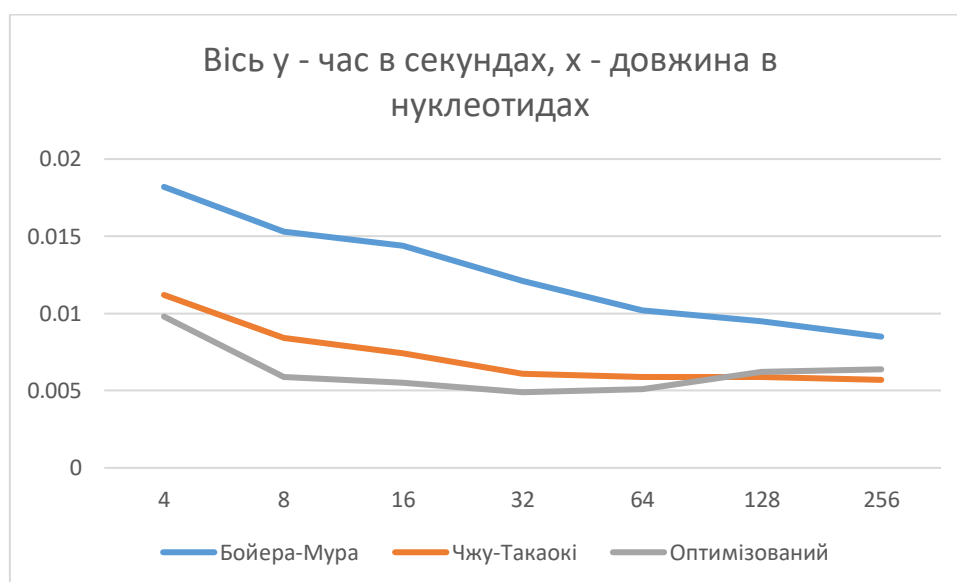
Рис. 4. Приклад роботи програми

Дані для аналізу були отримані після пошуку на геномі *E. Coli*. Результати роботи алгоритмів наведені в таблиці 1. Кожен алгоритм був протестований 50 разів на алфавіті в 4 символи (аденін, гуанін, цитозин, тимін) та на підрядках довжини в 4, 8, 16, 32, 64, 128 та 256 нуклеотидів. В таблиці 1 представлена інформація про часові показники роботи алгоритмів, що вимірювались у секундах. Також ці показники візуалізовані на графіку, для зручності алгоритм Райта відсутній на графіку (рис. 5).

Таблиця 1

Часові показники роботи алгоритмів

Довжина зразка Алгоритм	4	8	16	32	64	128	256
БМ	0,0182	0,0153	0,0144	0,0121	0,0102	0,0095	0,0085
Райта	0,0221	0,0242	0,0251	0,0254	0,0672	0,1267	0,2359
Чжу-Такаокі	0,0112	0,0084	0,0074	0,0061	0,0059	0,0059	0,0057
Оптимізований	0,0098	0,0059	0,0055	0,0049	0,0051	0,0062	0,0064

**Рис. 5. Часові показники роботи алгоритмів**

Отримані результати показують переваги та недоліки використання евристик алгоритму Райта в алгоритмі Чжу-Такаокі. Оптимізований алгоритм домінує на довжинах підрядка в 4, 8, 16, 32 та 64 нуклеотидів, але при збільшенні зразку до розміру в 128 нуклеотидів він починає здавати свої позиції. Звичайні алгоритми Чжу-Такаокі і Бойера-Мура більш стабільні на великих довжинах але алгоритм Райта, як і передбачалося, не може бути ефективно використаний для задач біоінформатики.

Висновки. У роботі були досліджені та проаналізовані 4 алгоритми пошуку підрядка в рядку. Оптимізований алгоритм Чжу-Такаокі ефективно працює на невеликих довжинах зразків (приблизно, до 100 символів), що може допомогти при діагностуванні однонуклеотидних поліморфізмів у біологічних послідовностях. Це зумовлено тим, що евристика порівняння першого та останнього символів втрачає ефективність на великих послідовностях. Серед розглянутих алгоритмів, для задач пошуку підрядка в рядку, довжина якого більше 100 символів, краще використовувати звичайний алгоритм Чжу-Такаокі.

Література

1. Айткулов П. Г. Обработка символьных массивов / П. Г. Айткулов // Управление большими системами: сборник трудов. 2010. №28. С. 126-178.
2. Солдатов Г. П., Татаринов А. А., Болдырихин Н. В. Основные алгоритмы поиска подстроки в строке / Солдатов Г. П., Татаринов А. А., Болдырихин Н. В // Academy. 2018. №5 (32). С. 8-10.
3. Желудков А. В., Макаров Д. В., Фадеев П. В. Исследование алгоритмов точного поиска подстроки в строке // Символ науки. 2016. №11-3. С. 61-62.
4. Boyer-Moore algorithm. URL: <http://www-igm.univ-mlv.fr/~lecroq/string/node14.html#SECTION00140>
5. Gusfield D. Algorithms on strings, trees, and sequences: computer science and computational biology / D. Gusfield. – New York, NY, USA : Cambridge University Press, 1997.
6. Сухачев А.И., Довгаль В.М., Гордиенко В.В. Способ поиска по образцу // Auditorium. 2016. №2(10). С. 57-66
7. Rahim R. Searching Process with Raita Algorithm and its Application / Rahim R. // Journal of Physics: Conference Series. Pub. 1, Vol. 1007, 2018.