|  |  |  |  |
| --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка  ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  **Кафедра програмних систем і технологій**  Дисципліна  **« Структури даних, аналіз і алгоритми  комп'ютерної обробки інформації »**  **Лабораторна робота № 1** | | | |
| **Виконав:** | Круць Михайло Ярославович | **Перевірив**: | Бичков Олексій Сергійович |
| Група | ІПЗ-12 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

**Умова задачі:**

Написати програму мовою C# з можливістю вибору різних алгоритмів пошуку. Продемонструвати роботу (ефективність, час виконання) програм на різних структурах даних (масив, лінійний зв’язаний список), з різними умовами, що забезпечують зменшення часу виконання. Навести аналіз отриманих результатів.

Реалізувати алгоритми:

* пошуку перебором елемента масиву, що дорівнює заданому значенню.
* пошуку з бар'єром елемента масиву, що дорівнює заданому значенню.
* бінарного пошуку елемента масиву рівного заданому значенню.
* бінарного пошуку елемента масиву, рівного заданому значенню, в якій нове значення індексу m визначалося б не як середнє значення між L і R, а згідно з правилом золотого перерізу.

**Аналіз задачі:**

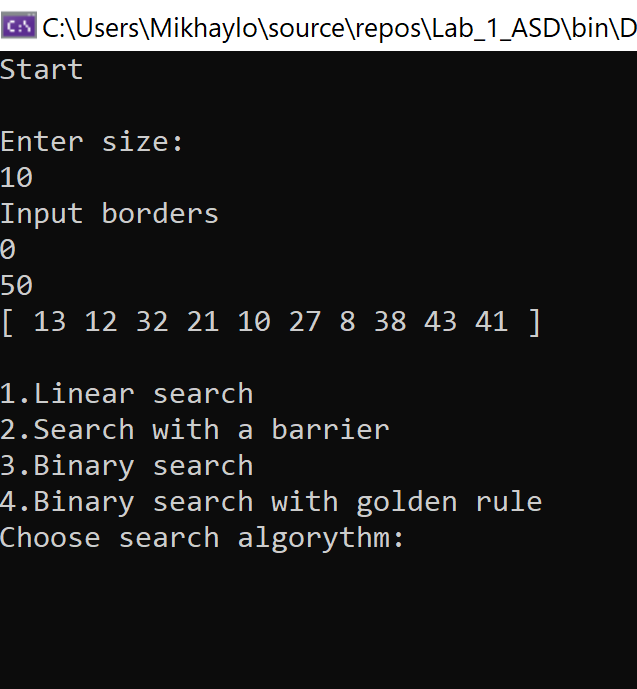
Програма повинна забезпечити меню вибору різних алгоритмів пошуку на різних структурах даних. Спочатку створюються зв’язний список та масив чисел. За допомогою циклу for заповнюються числами двоє типів даних. Числа визначаються за допомогою класу “Random”. Швидкість виконання програми визначається за допомогою класу “Stopwatch” Програма за допомогою окремих методів у класі «» створює масив та лінійний зв’язний список, розмірність яких задається на вибір користувача.

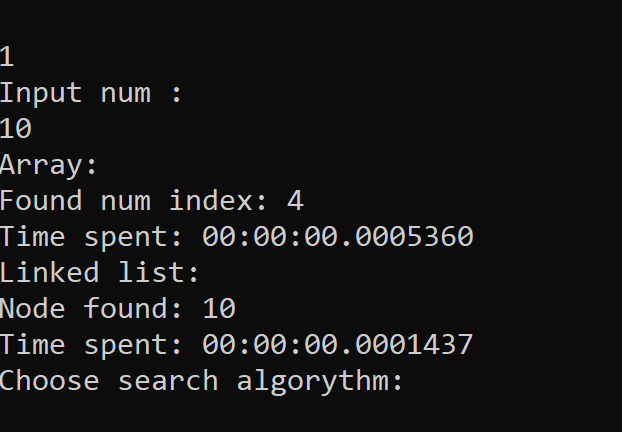
Принцип роботи алгоритмів для масиву:

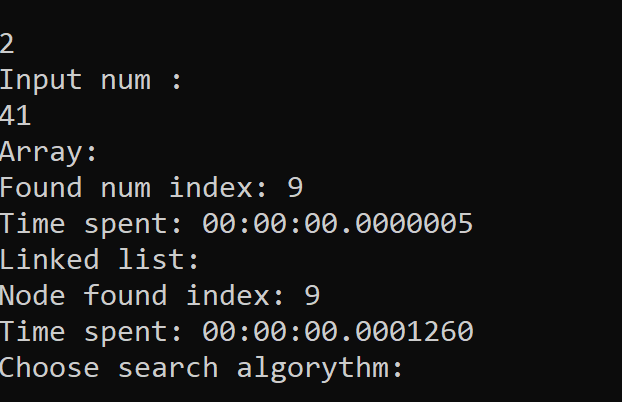
* Пошук перебором: Сутність алгоритму у послідовному порівнянні кожного елемента масиву із шуканим елементом.
* Пошук з бар’єром: Сутність алгоритму у додаванні у кінець масиву елемента із значенням шуканого. Тобто рано чи пізно співпадіння відбудеться 100%.
* Бінарний пошук: Спочатку масив відсортовується за зростанням. Потім знаходження середнього елемента масиву. Порівняння шуканого елемента із середнім елементом масиву. В залежності від результату порівняння відкидаємо ту чи іншу половину масиву змінюючи позицію початку чи кінця масиву.
* Бінарний пошук із зміненою умовою середнього елемента відповідно до золотого перерізу: Алгоритм виконується відповідно до бінарного пошуку але при знаходженні середнього елемента порівнюються розміри границь правилом ~68% на ~32%(золотий переріз).

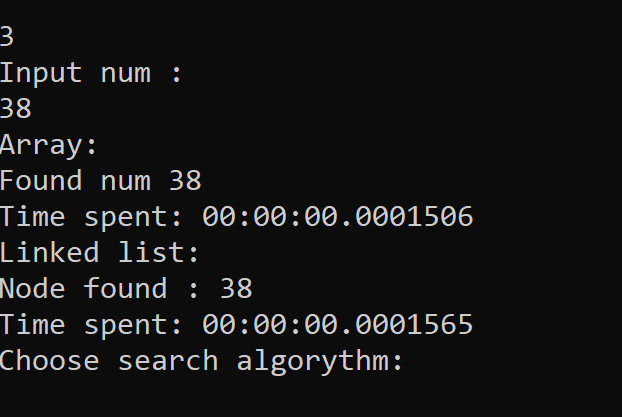
**Структура основних вхідних та вихідних даних:**

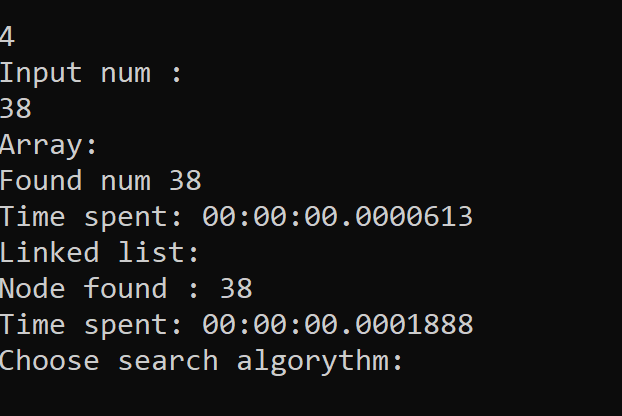
* Вхідні дані:
* Масив — це набір елементів подібного типу даних . До елементів масиву можна отримати доступ за допомогою індексу масиву. Елементи даних зберігаються в суміжних місцях в пам'яті. Операції над елементами як додавання та видалення потребують створення нового масиву. Пам'ять виділяється під час компіляції (статичний тип).Розмір масиву необхідно вказати під час оголошення/ініціалізації масиву.
* Лінійний зв’язний список — це впорядкований набір елементів одного типу, в якій кожен елемент поєднується з наступним за допомогою покажчиків. Доступ до елементів здійснюється послідовно. Нові елементи можна зберігати в будь-якому місці, а посилання на новий елемент створюється за допомогою покажчиків. Пам'ять виділяється під час виконання (динамічний тип). Розмір зв’язаного списку зростає/зменшується в міру додавання/видалення нових елементів.
* При запуску програми користувач вводить розмір структур даних, та вводить діапазон значень вхідних даних.
* Вихідні дані:
* Вивід створених чисел та меню вибору алгоритму пошуку:



* Користувач вибирає алгоритм та на виході отримує повідомлення про результат знайденного елемента та швидкість виконання алгоритму:

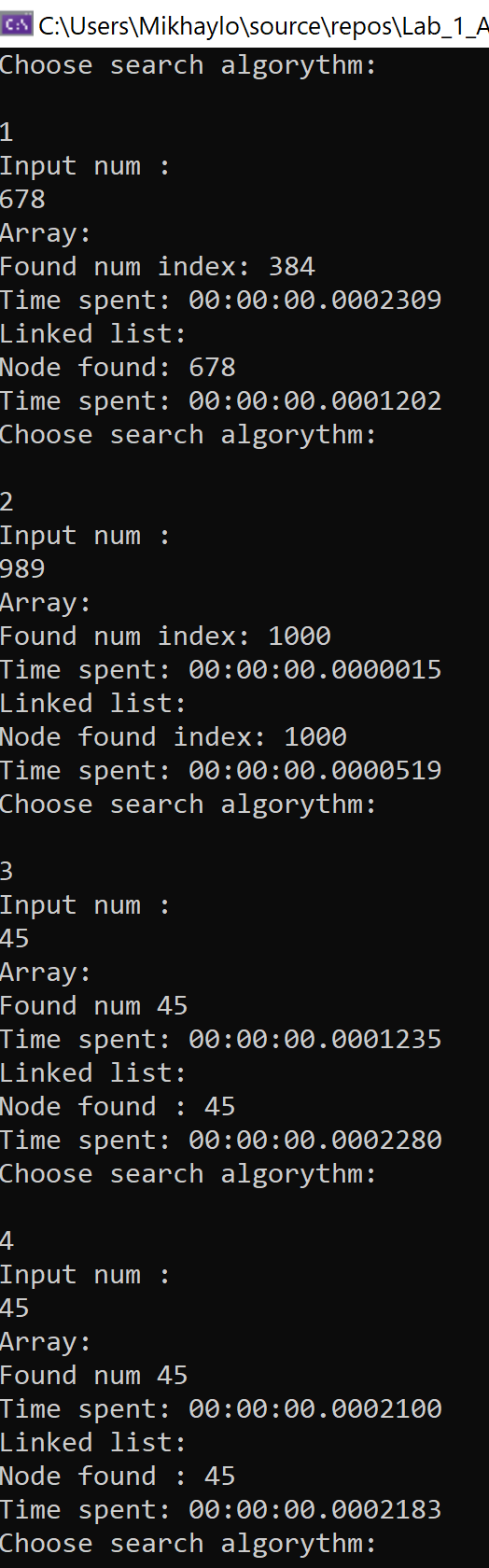




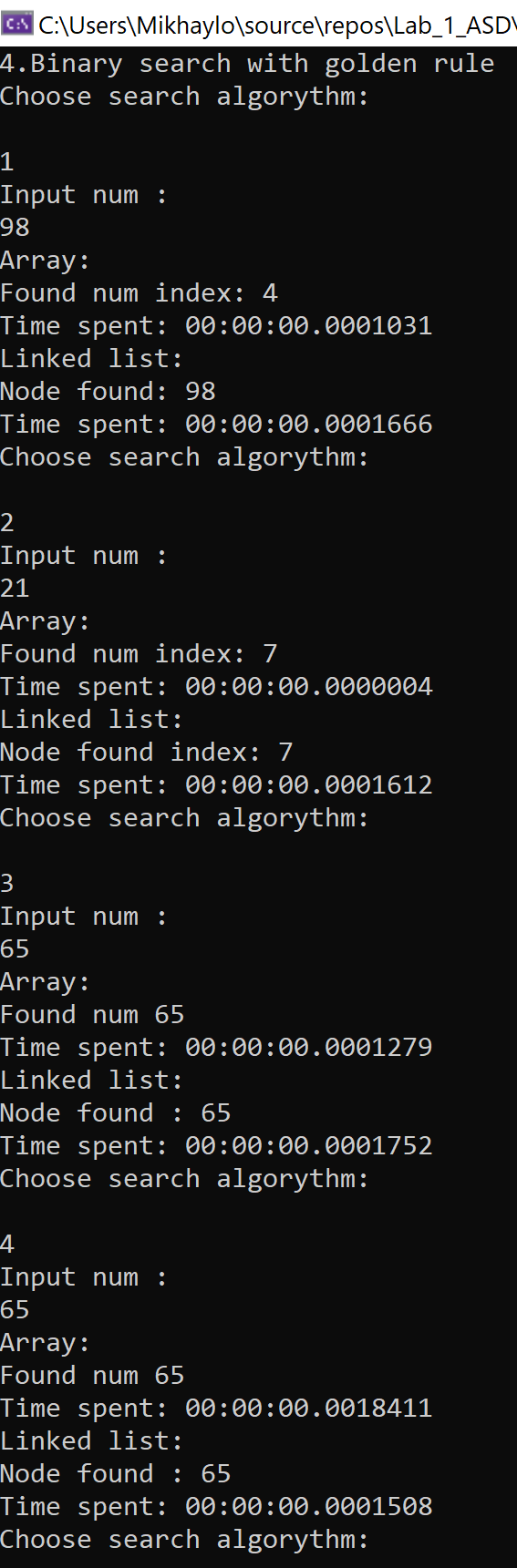


Набір тестів:

* Розмір масиву та зв’язного списку на 1000 елементів



* Розмір масиву та зв’язного списку на 20 елементів:



**Результати тестування та аналіз отриманних помилок:**

У результаті тестування виявлено, що методи пошуку за певними алгоритмами працюють коректно.. Найшвидшим у виконанні задачі виявився бар’ерний пошук у масиві. Якщо порівнювати структури даних то у результаті більшість алгоритмів пошуку виконуюються швидше із масивом, але у випадку бінарного пошуку із правилом золотого перерізу, швидшим виявився у виконанні лінійний зв’язний список.