Лабораторна робота № 2

Тема: Методи пошуку в одновимірних масивах.

Мета роботи: набуття практичних навичок застосування алгоритмів пошуку.

Хід роботи

Завдання 1. Реалізуйте модифікацію алгоритму бінарного пошуку з постійною перевіркою, чи шуканий елемент вже знайдено (це дозволить зменшити реальний час виконання і принесе вам додаткових пів балу до звіту ☺).

Лістинг 2.1 – Код програми 1

```
//Чавурська Єлизавета, ЛР2, Варіант 20, АіСД
#include <iostream>
using namespace std;
int main() {
	setlocale(LC_CTYPE, "Ukr");
	int arr[] = { 2, 5, 8, 12, 16, 23, 38, 56, 72, 91 };
	int n = 10;
	int target;
	cout << "Введіть шуканий елемент: ";
	cin >> target;
	int left = 0, right = n - 1;
	bool found = false;
	int mid;
	while (left <= right && !found) {
		mid = (left + right) / 2;
		if (arr[mid] == target) {
			found = true;
		}
		else if (arr[mid] < target) {
			left = mid + 1;
		}
		else {
			right = mid - 1;
		}
	}
	if (found)
		cout << "Елемент знайдено на позиції: " << mid << endl;
	else
		cout << "Елемент не знайдено!" << endl;
	cout << "Чавурська Єлизавета, ЛР2, Варіант 20, завдання 1" << endl;
	return 0;
}
```

```
Введ?ть шуканий елемент: 5
Елемент знайдено на позиц?ї: 1
Чавурська Єлизавета, ЛР2, Вар?ант 20, завдання 1
```

Рисунок 2.1 – Вікно результатів програми

```
Введ?ть шуканий елемент: 2
Елемент знайдено на позиц?ї: 0
Чавурська Єлизавета, ЛР2, Вар?ант 20, завдання 1
```

Рисунок 2.2 – Вікно результатів програми

Завдання 2. Модифікуйте алгоритм бінарного пошуку найпершого входження шуканого числа для пошуку останнього входження шуканого числа.

Лістинг 2.2 – Код програми 2

```
//Чавурська Єлизавета, ЛР2, Варіант 20, АіСД
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main() {
	setlocale(LC_CTYPE, "ukr");
	int n;
	cout << "Введіть кількість елементів масиву: ";
	cin >> n;
	vector<int> arr(n);
	cout << "Введіть елементи масиву (через пробіл): ";
	for (int i = 0; i < n; i++)
		cin >> arr[i];
	// Сортуємо масив
	sort(arr.begin(), arr.end());
	int target;
	cout << "Введіть шуканий елемент: ";
	cin >> target;

	int left = 0, right = n - 1;
	int result = -1;
	// Модифікований бінарний пошук останнього входження
	while (left <= right) {
		int mid = (left + right) / 2;
		if (arr[mid] == target) {
			result = mid;    // запам'ятовуємо позицію
			left = mid + 1;  // рухаємось праворуч, шукаємо останнє
входження
		}
		else if (arr[mid] < target) {
			left = mid + 1;
```

```
            }
            else {
                  right = mid - 1;
            }
      }
      cout << "\nВідсортований масив: ";
      for (int x : arr) cout << x << " ";
      cout << endl;
      if (result != -1)
            cout << "Останнє входження елемента " << target << " на позиції: "
<< result << endl;
      else
            cout << "Елемент не знайдено!" << endl;

      cout << "Чавурська Єлизавета, ЛР2, Варіант 20, завдання 2" << endl;
      return 0;
}
```

```
Введ?ть к?льк?сть елемент?в масиву: 3
Введ?ть елементи масиву (через проб?л): 34
20 1
Введ?ть шуканий елемент: 20

В?дсортований масив: 1 20 34
Останнє входження елемента 20 на позиц?ї: 1
Чавурська Єлизавета, ЛР2, Вар?ант 20, завдання 2
```

Рисунок 2.3 – Вікно результатів програми 2

```
Введ?ть к?льк?сть елемент?в масиву: 8
Введ?ть елементи масиву (через проб?л): 1 2 3 4 5 6 7 8
Введ?ть шуканий елемент: 0

В?дсортований масив: 1 2 3 4 5 6 7 8
Елемент не знайдено!
Чавурська Єлизавета, ЛР2, Вар?ант 20, завдання 2
```

Рисунок 2.4 – Вікно результатів програми 2

Завдання для виконання

1. Реалізувати функцію пошуку елементу в масиві, використовуючи лінійний метод, з не менш як трьома різними вхідними даними, у тому числі:

- елемент знаходиться на початку списку;

- елемент знаходиться в кінці списку; - елемент є приблизно в середині списку; - елемента не має у списку тощо.

Лістинг 2.3 – Код програми 3

```
//Чавурська Єлизавета, ЛР2, Варіант 20, АіСД
#include <iostream>
using namespace std;
// Лінійний пошук
int func1(int arr[], int size, int value) {
	int count = 0;
	for (int i = 0; i < size; i++) {
		if (arr[i] == value)
			count++;
	}
	return count;
}
// Функція для лінійного пошуку елемента
int func2(int arr[], int size, int value) {
	for (int i = 0; i < size; i++) {
		if (arr[i] == value)
			return i;
	}
	return -1;
}
int main() {
	setlocale(LC_CTYPE, "ukr");
	int n, m;
	cout << "Введіть кількість елементів масиву A: ";
	cin >> n;
	int A[100];
	cout << "Введіть елементи масиву A: ";
	for (int i = 0; i < n; i++)
		cin >> A[i];
	cout << "Введіть кількість елементів масиву B: ";
	cin >> m;
	int B[100];
	cout << "Введіть елементи масиву B: ";
	for (int i = 0; i < m; i++)
		cin >> B[i];
	cout << "\nЕлементи з масиву A, кратні 7, які тричі повторюються у
масиві B:\n";
	bool found = false;
	for (int i = 0; i < n; i++) {
		if (A[i] % 7 == 0) {
			int count = func1(B, m, A[i]);
			if (count == 3) {
				cout << A[i] << " ";
				found = true;
			}
		}
	}
```

```
    if (!found)
        cout << "Таких елементів немає.";
    cout << "\n\nПеревірка лінійного пошуку для різних випадків:\n";
    if (m > 0) {
        int test1 = B[0];       // початок
        int test2 = B[m / 2];   // середина
        int test3 = B[m - 1];   // кінець
        int test4 = 9999;       // відсутній
        int tests[] = { test1, test2, test3, test4 };
        const char* labels[] = { "початок", "середина", "кінець", "немає"
};
        for (int i = 0; i < 4; i++) {
            int pos = func2(B, m, tests[i]);
            if (pos != -1)
                cout << "Елемент (" << labels[i] << ") " << tests[i]
                << " знайдено на позиції " << pos << endl;
            else
                cout << "Елемент (" << labels[i] << ") " << tests[i]
                << " не знайдено у масиві B\n";
        }
    }
    cout << "Чавурська Єлизавета, ЛР2, Варіант 20" << endl;
    return 0;
}
```

```
Введ?ть к?льк?сть елемент?в масиву A: 10
Введ?ть елементи масиву A: 1 2 3 4 5 6 7 8 9 10
Введ?ть к?льк?сть елемент?в масиву B: 5
Введ?ть елементи масиву B: 1 2 3 4 5

Елементи з масиву A, кратн? 7, як? трич? повторюються у масив? B:
Таких елемент?в немає.

Перев?рка л?н?йного пошуку для р?зних випадк?в:
Елемент (початок) 1 знайдено на позиц?ї 0
Елемент (середина) 3 знайдено на позиц?ї 2
Елемент (к?нець) 5 знайдено на позиц?ї 4
Елемент (немає) 9999 не знайдено у масив? B
Чавурська Єлизавета, ЛР2, Вар?ант 20
```

Рисунок 2.5 – Вікно результатів програми 3

```
Введ?ть к?льк?сть елемент?в масиву A: 4
Введ?ть елементи масиву A: 1 2 3 4
Введ?ть к?льк?сть елемент?в масиву B: 2
Введ?ть елементи масиву B: 1 2

Елементи з масиву A, кратн? 7, як? трич? повторюються у масив? B:
Таких елемент?в немає.

Перев?рка л?н?йного пошуку для р?зних випадк?в:
Елемент (початок) 1 знайдено на позиц?ї 0
Елемент (середина) 2 знайдено на позиц?ї 1
Елемент (к?нець) 2 знайдено на позиц?ї 1
Елемент (немає) 9999 не знайдено у масив? B
Чавурська Єлизавета, ЛР2, Вар?ант 20
```

Рисунок 2.6 – Вікно результатів програми 3

2. Реалізувати функцію пошуку елементу в масиві, використовуючи двійковий метод з не менш як трьома різними вхідними даними. Результати занести у звіт

Лістинг 2.4 – Код програми 4

```
//Чавурська Єлизавета, ЛР2, Варіант 20, АіСД
#include <iostream>
using namespace std;
// Функція двійкового пошуку
int binarySearch(int arr[], int size, int value) {
	int left = 0;
	int right = size - 1;
	while (left <= right) {
		int mid = (left + right) / 2;
		if (arr[mid] == value)
			return mid; // знайдено
		if (arr[mid] < value)
			left = mid + 1;
		else
			right = mid - 1;
	}
	return -1;
}
int main() {
	setlocale(LC_CTYPE, "ukr");
	int n;
	cout << "Введіть кількість елементів масиву: ";
	cin >> n;
	int A[100];
	cout << "Введіть елементи масиву у зростаючому порядку:\n";
	for (int i = 0; i < n; i++)
		cin >> A[i];
```

```
    cout << "\nМасив зчитано. Перевіримо пошук для різних випадків:\n";
    int test1 = A[0];
    int test2 = A[n / 2];
    int test3 = 9999;
    int tests[] = { test1, test2, test3 };
    const char* labels[] = { "початок", "середина", "немає" };
    for (int i = 0; i < 3; i++) {
        int pos = binarySearch(A, n, tests[i]);
        if (pos != -1)
            cout << "Елемент (" << labels[i] << ") " << tests[i]
            << " знайдено на позиції " << pos << endl;
        else
            cout << "Елемент (" << labels[i] << ") " << tests[i]
            << " не знайдено у масиві.\n";
    }
    cout << "Чавурська Єлизавета, ЛР2, Варіант 20" << endl;
    return 0;
}
```

```
Введ?ть к?льк?сть елемент?в масиву: 4
Введ?ть елементи масиву у зростаючому порядку:
1 2 3 4

Масив зчитано. Перев?римо пошук для р?зних випадк?в:
Елемент (початок) 1 знайдено на позиц?ї 0
Елемент (середина) 3 знайдено на позиц?ї 2
Елемент (немає) 9999 не знайдено у масив?.
Чавурська Єлизавета, ЛР2, Вар?ант 20
```

Рисунок 2.7 – Вікно результатів програми 4

```
Введ?ть к?льк?сть елемент?в масиву: 6
Введ?ть елементи масиву у зростаючому порядку:
1 2 3 4 56 6

Масив зчитано. Перев?римо пошук для р?зних випадк?в:
Елемент (початок) 1 знайдено на позиц?ї 0
Елемент (середина) 4 знайдено на позиц?ї 3
Елемент (немає) 9999 не знайдено у масив?.
Чавурська Єлизавета, ЛР2, Вар?ант 20
```

Рисунок 2.8 – Вікно результатів програми 4

3. Зробити висновок про роботу та записати у звіт.

Висновок: на цій лабораторній роботі я набула практичних навичок застосування алгоритмів пошуку.