

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 3

На тему: «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання.
Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки
та робота з масивами та структурами.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 4
ВНС Лабораторної Роботи № 5
Алготестер Лабораторної Роботи №2
Алготестер Лабораторної Роботи №3
Практичних Робіт до блоку №4

Виконав:

Студент групи ШІ-11
Оренчук Юрій Миколайович

Львів 2024

Тема: Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

Мета: Навчитися використовувати одновимірні та двовимірні масиви, вивчити основи вказівників та посилань, опанувати структурами даних та вкладеними структурами, освоїти динамічні масиви та класи пам'яті.

Теоретичні відомості:

Тема №1: Одновимірні масиви, двовимірні масиви, вказівники та посилання, структури даних, алгоритм обробки та робота з масивами та структурами

Тема №2: Вкладені структури

Тема №3: Динамічні масиви

Індивідуальний план опрацювання теорії:

Тема №1: <https://www.youtube.com/watch?v=-TkoO8Z07hI>

Тема №2: <https://www.youtube.com/watch?v=Ldsj83BsKo4>

Тема №3: <https://www.youtube.com/watch?v=Ny3zrODxzTs>

Виконання роботи:

Завдання №1: VNS Lab 4 Variant 5

- 1) Сформуванати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити елементи кратні 7.
- 4) Додати після кожного непарного елемента масиву елемент зі значенням 0.
- 5) Роздрукувати отриманий масив.

Завдання №2: VNS Lab 5 Variant 24

Використовуючи функції, розв'язати зазначене у варіанті завдання.
Масив

повинен передаватися у функцію як параметр.

Визначити скільки елементів двовимірного масиву більші від будь-якого елемента на головній діагоналі.

Завдання №3: Algotester Lab 2 Variant 1

Lab 2v1

Limits: 1 sec., 256 MiB

У вас є дорога, яка виглядає як N чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елемента.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

Input

У першому рядку ціле число N - кількість чисел

У другому рядку масив r , який складається з N цілих чисел

Output

Єдине ціле число m - мінімальна втома, яку можна отримати

Constraints

$$1 \leq N \leq 10^5$$

$$0 \leq r_i \leq 10^5$$

Завдання №4: Algotester Lab 3 Variant 2

Lab 3v2

Limits: 1 sec., 256 MiB

Вам дано 2 масиви розміром N та M . Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

Input

У першому рядку ціле число N

у другому рядку N цілих чисел $a_1 \dots a_n$

У третьому рядку ціле число M

у четвертому рядку M цілих чисел $b_1 \dots b_n$

Output

У першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об'єднанням двох даних).

Constraints

$$0 \leq N \leq 100$$

$$0 \leq a_i \leq 100$$

$$0 \leq M \leq 100$$

$$0 \leq b_i \leq 100$$

Завдання №5: Class Practice Work

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

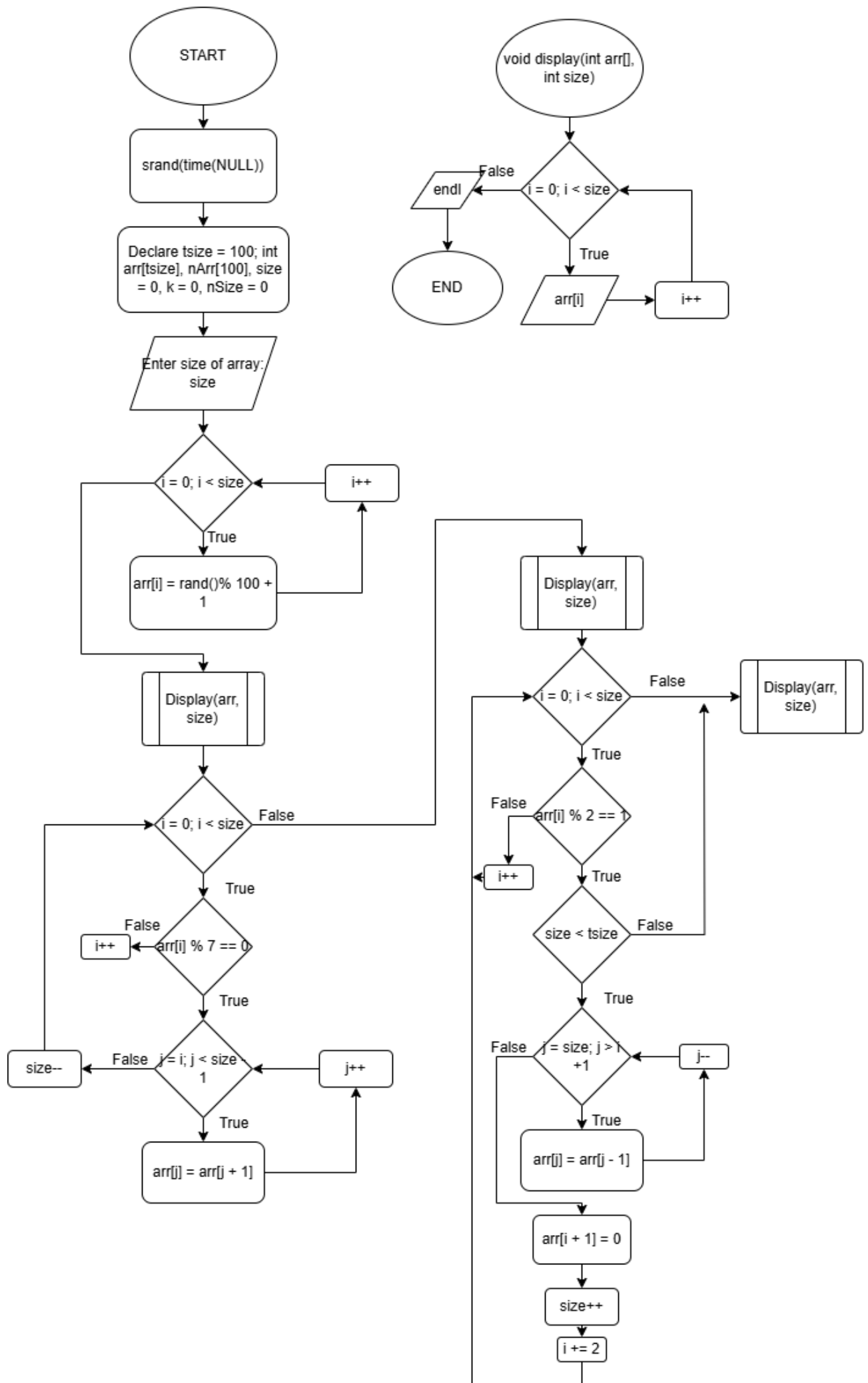
Вимоги:

1. Визначення функції:
 - a. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
 - a. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
 - a. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - b. *bool isPalindrome(ціле число);*
4. Рекурсія:
 - a. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

Дизайн та планована оцінка часу виконання завдань:

Завдання №1: VNS Lab 4 Variant 5

Планований час: 1 год



Завдання №2: VNS Lab 5 Variant 24

Планований час: 45 хв

Завдання №3: Algotester Lab 2 Variant 1

Планований час: 45 хв

Завдання №4: Algotester Lab 3 Variant 2

Планований час: 30 хв

Завдання №5: Class Practice Work

Планований час: 1 год

Код програм з посиланням на зовнішні ресурси:

Завдання №1: VNS Lab 4 Variant 5

```
30     for (int i = 0; i < size;){
31         if (arr[i] % 7 == 0) {
32             for (int j = i; j < size - 1; j++){
33                 arr[j] = arr[j + 1];
34             }
35             size--;
36         } else {
37             i++;
38         }
39     }
40
41     Display(arr, size);
42
43     for (int i = 0; i < size;){
44         if (arr[i] % 2 == 1){
45             if (size < tsize){
46                 for (int j = size; j > i + 1; j--){
47                     arr[j] = arr[j - 1];
48                 }
49                 arr[i + 1] = 0;
50                 size++;
51                 i += 2;
52             } else break;
53         } else i++;
54     }
55
56     Display(arr, size);
57
58     return 0;
59 }
```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_yurii_orenchuk/ai_11/yurii_orenchuk/epic_4/vns_lab_4_task_1_variant_5_yurii_orenchuk.cpp

Завдання №2: VNS Lab 5 Variant 24

```
1  #include <iostream>
2  using namespace std;
3
4  int max_diag(int (&matrix)[100][100], const int &n) {
5      int max = 0;
6      for (int k = 0; k < n; k++) {
7          if (max < matrix[k][k])
8              max = matrix[k][k];
9      }
10     return max;
11 }
12
13 int main() {
14     int n;
15     cout << "Введіть величину матриці: ";
16     cin >> n;
17
18     if (n > 100) {
19         cout << "Занадто великий масив, спробуйте ще раз";
20         return 1;
21     }
22
23     int matrix[100][100];
24
25     cout << "Введіть елементи цієї матриці по рядках: ";
26     for (int i = 0; i < n; i++) {
27         for (int j = 0; j < n; j++) {
28             cin >> matrix[i][j];
29         }
30     }
31
32     int MAX = max_diag(matrix, n);
33     int counter = 0;
34
35     for (int i = 0; i < n; i++) {
36         for (int j = 0; j < n; j++) {
37             if (matrix[i][j] > MAX) counter++;
38         }
39     }
40
41     cout << "Кількість елементів множини, які більші за будь який елемент діагоналі: " << counter;
42
43     return 0;
44 }
```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_yurii_orenychuk/ai_11/yurii_orenychuk/epic_4/vns_lab_5_task_1_variant_24_yurii_orenychuk.cpp

Завдання №3: Algotester Lab 2 Variant 1

```
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4  using namespace std;
5
6  struct Temporary{
7      int tmp_max;
8      int tmp_min;
9      vector<int> temp;
10 };
11
12 struct Info{
13     int max_val;
14     int min_val;
15     int m;
16     Temporary temp;
17 };
18
19 int main(){
20     Info info;
21     int n;
22     cin >> n;
23     if(n < 1) return 0;
24
25     vector<int> r(n);
26     for(int i = 0; i < n; i++){
27         cin >> r[i];
28     }
```

```
30     info.max_val = *max_element(r.begin(), r.end());
31     info.min_val = *min_element(r.begin(), r.end());
32     info.m = info.max_val - info.min_val;
33
34     for (int i = 0; i < n; ++i) {
35         info.temp.tmp_max = info.max_val;
36         info.temp.tmp_min = info.min_val;
37
38         if (r[i] == info.max_val || r[i] == info.min_val) {
39             info.temp.temp = r;
40             info.temp.temp.erase(info.temp.temp.begin() + i);
41             info.temp.tmp_max = *max_element(info.temp.temp.begin(), info.temp.temp.end());
42             info.temp.tmp_min = *min_element(info.temp.temp.begin(), info.temp.temp.end());
43         }
44
45         if((info.temp.tmp_max - info.temp.tmp_min) < info.m)
46             info.m = info.temp.tmp_max - info.temp.tmp_min;
47     }
48
49     cout << info.m << endl;
50
51     return 0;
52 }
```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_yurii_orenchuk/ai_11/yurii_orenchuk/epic_4/algotester_lab_2_variant_1_yurii_orenchuk.cpp

Завдання №4: Algotester Lab 3 Variant 2

```
24     for (int i = 0; i < N; i++) {
25         for (int j = 0; j < M; j++) {
26             if (a[i] == b[j]) {
27                 cnt1++;
28                 cnt2--;
29                 break;
30             }
31         }
32     }
33
34     cout << cnt1 << endl;
35     cout << cnt2;
36
37     delete[] a;
38     delete[] b;
39
40     return 0;
41 }
```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_yurii_orenchuk/ai_11/yurii_orenchuk/epic_4/algotester_lab_3_variant_2_yurii_orenchuk.cpp

Завдання №5: Class Practice Work

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  bool isPalindrome(string str, int start, int end){
6      if(start >= end)
7          return true;
8
9      if (str[start] != str[end])
10         return false;
11
12         return isPalindrome(str, ++start, --end);
13     }
14
15     bool isPalindrome(int num) {
16         if (num < 0) return false;
17
18         int origin = num;
19         int rev = 0;
20
21         while (num > 0) {
22             int digit = num % 10;
23             rev = rev * 10 + digit;
24             num /= 10;
25         }
26
27         return origin == rev;
28     }
29
30
```

```

31  int main(){
32      string word;
33      int number;
34
35      cout << "Enter a word: ";
36      getline(cin, word);
37
38      int start = 0;
39      int end = word.size()-1;
40
41      if (isPalindrome(word, start, end)){
42          cout << "This word is a palindrome\n";
43      }else cout << "This word is not a palindrome\n";
44
45
46      cout << "Enter a number: ";
47      cin >> number;
48      if (isPalindrome(number)){
49          cout << "This number is a palindrome";
50      }else cout << "This number is not a palindrome";
51
52
53      return 0;
54  }

```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_4_practice_and_labs_yurii_orenchuk/ai_11/yurii_orenchuk/epic_4/practice_work_task_1_yurii_orenchuk.cpp

Результат виконання завдань, тестування та фактично затрачений час:

Завдання №1: VNS Lab 4 Variant 5

Фактично затрачений час: 45 хв

```

Enter size of array: 20
84 71 85 51 79 46 54 80 81 85 47 5 43 63 53 76 19 92 67 41
71 85 51 79 46 54 80 81 85 47 5 43 53 76 19 92 67 41
71 0 85 0 51 0 79 0 46 54 80 81 0 85 0 47 0 5 0 43 0 53 0 76 19 0 92 67 0 41 0

```

Завдання №2: VNS Lab 5 Variant 24

Фактично затрачений час: 1 год

```
Введіть величину матриці: 4
Введіть елементи цієї матриці по рядках: 1
6
7
3
8
4
1
0
16
2
3
9
5
2
9
4
Кількість елементів множини, які більші за будь який елемент діагоналі: 7
PS C:\Users\admin1\Documents\CPP>
```

Завдання №3: Algotester Lab 2 Variant 1

Фактично затрачений час: 45 хв

```
5
1
2
2
2
4
1
PS C:\Users\admin1\Documents\CPP>
```

Завдання №4: Algotester Lab 3 Variant 2

Фактично затрачений час: 30 хв

```
5
1
2
3
4
5
5
4
5
6
7
8
2
8
PS C:\Users\admin1\Documents\CPP>
```

Завдання №5: Class Practice Work

Фактично затрачений час: 1,5 год

```
Enter a word: hannah
This word is a palindrome
Enter a number: 19091
This number is a palindrome
PS C:\Users\admin1\Documents\CPP>
```

Висновки: Я навчився використовувати одновимірні та двовимірні масиви, вивчив основи вказівників та посилань, опанував структурами даних та вкладеними структурами, освоїв динамічні масиви та класи пам'яті.

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/346