

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



Звіт

про виконання лабораторних та практичних робіт блоку № 4
про виконання лабораторних та практичних робіт блоку № 4

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-13
Шийка Стефан Андрійович

Львів 2024

Тема роботи:

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

Мета роботи:

Дослідження одновимірних і двовимірних масивів для зберігання і впорядкування даних, що забезпечує швидкий доступ і обробку великих обсягів інформації.

Дослідження вказівників та посилань для розуміння адресації пам'яті та оптимізації використання ресурсів, що дозволяє ефективніше працювати з динамічними структурами даних.

Дослідження динамічних масивів для створення програм із змінною кількістю елементів, що підвищує гнучкість і адаптивність коду.

Дослідження структур даних та вкладених структур для організації складних об'єктів, що забезпечує кращу структуру і читабельність програмного коду.

Дослідження алгоритмів обробки масивів і структур для реалізації ефективної обробки даних, що сприяє написанню оптимізованих і масштабованих програм.

Теоретичні відомості:

У даній роботі розглядаються основні принципи роботи з масивами та структурами даних, зокрема одновимірні й двовимірні масиви для організації і зберігання великих обсягів даних. Особливу увагу приділено вказівникам і посиланням як засобам управління пам'яттю та ефективного доступу до даних. Розглянуто динамічні масиви, які забезпечують гнучке управління розміром даних під час виконання програми. Досліджено основи структур даних і вкладених структур для створення складних, логічно організованих об'єктів. Описано алгоритми обробки масивів і структур, що дозволяють ефективно виконувати операції пошуку, сортування і модифікації даних, покращуючи оптимізацію коду.

Джерела:

-aCode

- Harvard CS50 lectures+tasks

Lab# programming: VNS Lab 4

18.

- 1) Реалізувати з використанням масиву однонаправлене кільце (перегляд можливий зліва направо, від останнього елемента можна перейти до першого).
- 2) Роздрукувати отриманий масив, починаючи з K-ого елемента і до K-1.
- 3) Додати в кільце перший і останній елементи.
- 4) Знищити з кільця парні елементи.
- 5) Роздрукувати отриманий масив, починаючи з K-ого елемента і до K-1.

```
1  #include<iostream>
2  using namespace std;
3
4  const int MAX = 100;
5  int arr[MAX] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
6  int arrSize = 10;
7  int k = 3;
8
9  int main() {
10     cout << "Array rotated by " << k << " positions:\n";
11     for (int i = k; i < arrSize + k; i++) {
12         cout << arr[i % arrSize] << " ";
13     }
14     cout << endl;
15
16     int first = -1;
17     int last = 10;
18     arrSize += 2;
19     arr[arrSize - 1] = last;
20
21     for (int i = arrSize - 3; i >= 0; i--) {
22         arr[i + 1] = arr[i];
23     }
24     arr[0] = first;
25
26     int counter = 0;
27     for (int i = 0; i < arrSize; i++) {
28         if (arr[i] % 2 == 0) {
29             counter++;
30         } else {
31             arr[i - counter] = arr[i];
32         }
33     }
34     arrSize -= counter;
35
36     cout << "Array rotated by " << k << " positions after adding new first and last elements and deleting the even ones:\n";
37     for (int i = k; i < arrSize + k; i++) {
38         cout << arr[i % arrSize] << " ";
39     }
40     cout << endl;
41 }
```

Array rotated by 3 positions:

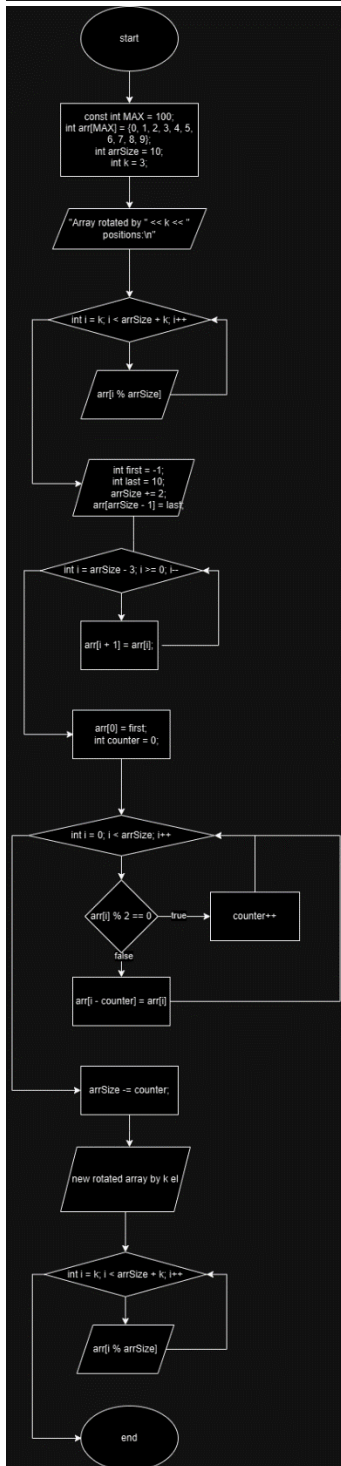
3 4 5 6 7 8 9 0 1 2

3 4 5 6 7 8 9 0 1 2

Array rotated by 3 positions after adding new first and last elements and deleting the even ones:

5 7 9 -1 1 3

PS C:\Users\user\Desktop\c++\epic4> □



Lab# programming: VNS Lab 5

18. Задано двовимірний масив $N \times N$. Послідовно розглядаються квадратні підмасиви, правий верхній елемент яких лежить на бічній діагоналі. У кожному такому підмасиві перебуває максимальний елемент. Шляхом перестановок рядків і стовпців (повністю) елемент треба перемістити в правий верхній кут підмасиву. Перевірити чи вийшла на бічній діагоналі спадаюча послідовність елементів.

```

1  #include <iostream>
2  using namespace std;
3
4  const int N = 4;
5
6  void rearrangeMatrix(int matrix[N][N]);
7  void moveMaxToTopRight(int matrix[N][N], int startRow, int startCol, int size);
8  void swapRows(int matrix[N][N], int row1, int row2);
9  void swapColumns(int matrix[N][N], int col1, int col2);
10 bool checkDiagonal(int matrix[N][N]);
11
12 int main(){
13     int matrix[N][N] = {
14         {12, 8, 3, 5},
15         {7, 15, 9, 10},
16         {2, 5, 18, 4},
17         {6, 11, 14, 20}
18     };
19
20     rearrangeMatrix(matrix);
21     bool result = checkDiagonal(matrix);
22
23     cout << "Matrix after rearrangement:\n";
24     for(int i = 0; i < N; i++){
25         for(int j = 0; j < N; j++){
26             cout << matrix[i][j] << " ";
27         }
28         cout << endl;
29     }
30
31     cout << "\nIs the anti-diagonal descending: " << (result ? "Yes" : "No") << endl;
32 }
33
34 void rearrangeMatrix(int matrix[N][N]){
35     for(int i = 0; i < N; i++){
36         moveMaxToTopRight(matrix, i, N - 1 - i, N - i);
37     }
38 }
39
40 void moveMaxToTopRight(int matrix[N][N], int startRow, int startCol, int size){
41     int maxVal = matrix[startRow][startCol];
42     int maxRow = startRow;
43     int maxCol = startCol;
44
45     for (int i = startRow; i < startRow + size; ++i) {
46         for (int j = startCol; j < startCol + size; ++j) {
47             if (matrix[i][j] > maxVal) {
48                 maxVal = matrix[i][j];
49                 maxRow = i;
50                 maxCol = j;
51             }
52         }
53     }
54
55     if (maxRow != startRow) {
56         swapRows(matrix, maxRow, startRow);
57     }
58     if (maxCol != startCol + size - 1) {
59         swapColumns(matrix, maxCol, startCol + size - 1);
60     }
61 }
62
63 void swapRows(int matrix[N][N], int row1, int row2) {
64     for (int col = 0; col < N; ++col) {
65         swap(matrix[row1][col], matrix[row2][col]);
66     }
67 }
68
69 void swapColumns(int matrix[N][N], int col1, int col2) {
70     for (int row = 0; row < N; ++row) {
71         swap(matrix[row][col1], matrix[row][col2]);
72     }
73 }
74
75 bool checkDiagonal(int matrix[N][N]){
76     for(int i = 0; i < N - 1; i++){
77         if(matrix[i][N - 1 - i] <= matrix[i + 1][N - 2 - i]) return false;
78     }
79     return true;
80 }
81
82

```

```
Matrix after rearrangement:
6 2 11 20
14 7 5 4
18 12 15 10
9 0 8 5

Is the anti-diagonal descending: No
```

Lab# programming: Algotester Lab 2

```
1  #include <iostream>
2  #include <climits>
3
4  using namespace std;
5
6  int main(){
7      int N, m;
8
9      cin >> N;
10     if(N <= 2){
11         cout << 0;
12         return 0;
13     }
14     int r[N];
15
16     for(int i = 0; i < N; i++){
17         cin >> r[i];
18     }
19
20
21     int min1 = INT_MAX, min2 = INT_MAX;
22     int max1 = INT_MIN, max2 = INT_MIN;
23
24     for(int i = 0; i < N; i++){
25         if (r[i] < min1) {
26             min2 = min1;
27             min1 = r[i];
28         } else if (r[i] < min2) {
29             min2 = r[i];
30         }
31
32         if (r[i] > max1) {
33             max2 = max1;
34             max1 = r[i];
35         } else if (r[i] > max2) {
36             max2 = r[i];
37         }
38     }
39
40     if(max1 - min2 <= max2 - min1){
41         m = max1 - min2;
42     }else{
43         m = max2 - min1;
44     }
45
46     cout << m;
47 }
```

a few seconds ago

Lab 2v1 - Lab 2v1

C++ 23

Accepted

0.003

1.234

1857424

Lab# programming: Algotester Lab 3

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main() {
7      string S;
8      getline(cin, S);
9      int len = S.length();
10     string result;
11
12     for (int i = 0; i < len; i++) {
13         int counter = 1;
14
15         while (i + 1 < len && S[i] == S[i + 1]) {
16             counter++;
17             i++;
18         }
19
20         result.push_back(S[i]);
21
22         if (counter > 1) {
23             result += to_string(counter);
24         }
25     }
26
27     cout << result;
28 }
```

a few seconds ago

Lab 3v3 - Lab 3v3

C++ 23

Accepted

0.003

1.254 1857430

Practice# programming: Class Practice Task

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  bool isPalindrome(const string& s, int start, int end){
7      if(start >= end) return true;
8      if(s[start] != s[end]) return false;
9      return isPalindrome(s, start+1, end-1);
10 }
11
12 bool isPalindrome(int n) {
13     if (n < 0) return false;
14
15     int original = n;
16     int reversed = 0;
17
18     while (n > 0) {
19         int digit = n % 10;
20         reversed = reversed * 10 + digit;
21         n /= 10;
22     }
23
24     return original == reversed;
25 }
26
27 int main(){
28     string s;
29     int n;
30     cout << "Input a string: ";
31     getline(cin, s);
32     cout << endl;
33     cout << "Input an integer: ";
34     cin >> n;
35     cout << endl;
36
37     cout << "String is palindrome: " << (isPalindrome(s, 0, s.length()-1) ? "Yes" : "No") << endl;
38     cout << "Integer is palindrome: " << (isPalindrome(n) ? "Yes" : "No") << endl;
39 }
```



```
Input a string: radar
Input an integer: 112211
String is palindrome: Yes
Integer is palindrome: Yes
```

```
Input a string: monkey
Input an integer: 12343
String is palindrome: No
Integer is palindrome: No
```

Practice# programming: Self Practice Task

```
1  ✓ #include <iostream>
2    #include <string>
3
4    using namespace std;
5
6  ✓ int main(){
7      int k;
8      string substr = "TOILET";
9      string s;
10     cin >> k;
11     cin.ignore();
12     getline(cin, s);
13
14     int len = s.length();
15     int counter = 0;
16     int position = 0;
17
18  ✓ while((position = s.find(substr, position)) != string::npos){
19     counter++;
20     position += substr.length();
21 }
22
23     cout << ((k <= counter) ? "YES" : "NO");
24 }
```

```
2
HELPTOILETMENPLEASETOILET
YES
PS C:\Users\user\Desktop\c++\epic4>
```

2 minutes ago	0081 - Допоможі чи заб'є?	C++ 23	Accepted	0.005	1.168	1857548
---------------	---------------------------	--------	----------	-------	-------	---------

PULL

Висновок: В процесі виконання лабораторної роботи я навчився використовувати одновимірні та двовимірні масиви для зберігання і впорядкування даних, що покращує доступ до великого обсягу інформації. Також я ознайомився з поняттями вказівників і посилань, що дозволяє

ефективно управляти пам'яттю і використовувати динамічні масиви. Окрім того, я досліджував структури даних та алгоритми обробки масивів, що сприяє написанню оптимізованих і масштабованих програм.