

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



Звіт

про виконання лабораторних та практичних робіт блоку № 4

На тему: «Програмування: алгоритм, програма, код. Системи числення.
Двійкова система числення. Розробка та середовище розробки програми.»

з дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 4

Виконав:
Студент групи ШІ-11
Силіч Богдан

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: зрозуміти основи роботи з одновимірними та двовимірними масивами, а також освоїти концепції вказівників, динамічних масивів і структур даних, що дозволить ефективно реалізовувати алгоритми обробки даних та структури, щоб спростити вирішення складних завдань.

Теоретичні відомості:

1. Класи пам'яті у C++
 - Статична пам'ять.
 - Динамічна пам'ять.
 - Поняття стеку.
 - Виділення та вивільнення пам'яті.
2. Вступ до Масивів і Вказівників:
 - Основи масивів: визначення, важливість, приклади використання.
 - Різниця між статичними та динамічними масивами.
 - Основи вказівників: що це таке, як вони працюють.
 - Взаємозв'язок між масивами та вказівниками.
 - Вступ до посилань: основні концепції та відмінності від вказівників.
3. Одновимірні Масиви:
 - Створення та ініціалізація одновимірних масивів.
 - Основні операції: індексація, присвоєння, читання.
 - Цикли та обхід масивів.
 - Використання функцій для роботи з масивами.
 - Приклади алгоритмів сортування та пошуку.
4. Вказівники та Посилання:
 - Використання вказівників для доступу до елементів масиву.
 - Арифметика вказівників.

- Різниця між вказівниками та посиланнями в контексті функцій.
 - Динамічне виділення пам'яті з використанням вказівників.
 - Використання вказівників для створення складних структур даних.
5. Двовимірні Масиви:
- Оголошення та ініціалізація двовимірних масивів.
 - Вкладені цикли для обходу двовимірних масивів.
 - Практичні приклади використання двовимірних масивів.
 - Передача двовимірних масивів у функції.
 - Застосування двовимірних масивів для розв'язання задач.
6. Динамічні Масиви:
- Основи динамічного виділення пам'яті.
 - Створення та управління динамічними масивами.
 - Використання операторів `new` та `delete` для управління пам'яттю.
 - Реалізація змінної розмірності масивів.
 - Передача динамічних масивів у функції.
7. Структури Даних:
- Оголошення та використання структур.
 - Використання масивів та вказівників у структурах.
 - Функції для обробки даних у структурах.

Індивідуальний план опрацювання теорії:

Класи пам'яті у C++

Вступ до Масивів і Вказівників

Одновимірні Масиви

Вказівники та Посилання

Двовимірні Масиви

Динамічні Масиви

Структури Даних

Джерела:

- Chat gpt
- Список відтворення на YouTube (
<https://youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&si=sXvmPdnGkwvJLXUi>)
- Власний досвід
- Лекції та практичні

Виконання роботи:

VNS Lab 4 Task 1 variant 6

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити елемент із заданим номером.
- 4) Додати після першого парного елемента масиву елемент зі значенням $M[I-1]+2$.
- 5) Роздрукувати отриманий масив.

VNS Lab 5 Task 1 variant 6

Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку й найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку й найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові точки.

Algotester Lab 2 variant 3

Вам дано масив цілих чисел розміром NN , на першій та останній клітинці розміщено по дрону.
Вони одночасно взлітають.

На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться.

Тобто лівий дрон у першу секунду з клітинки з індексом 11 перелетить у клітинку з індексом a_{11} , тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якщо 2 дрони опиняються в одній клітинці - ви виводите **Collision**.

Якщо лівий дрон опиниться справа від правого - це **Miss**

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках a_i та a_{i+1} - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

Algotester Lab 3 variant 6

Вам дана стрічка s .

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

Class Practice Task

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

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

Self Practice Task

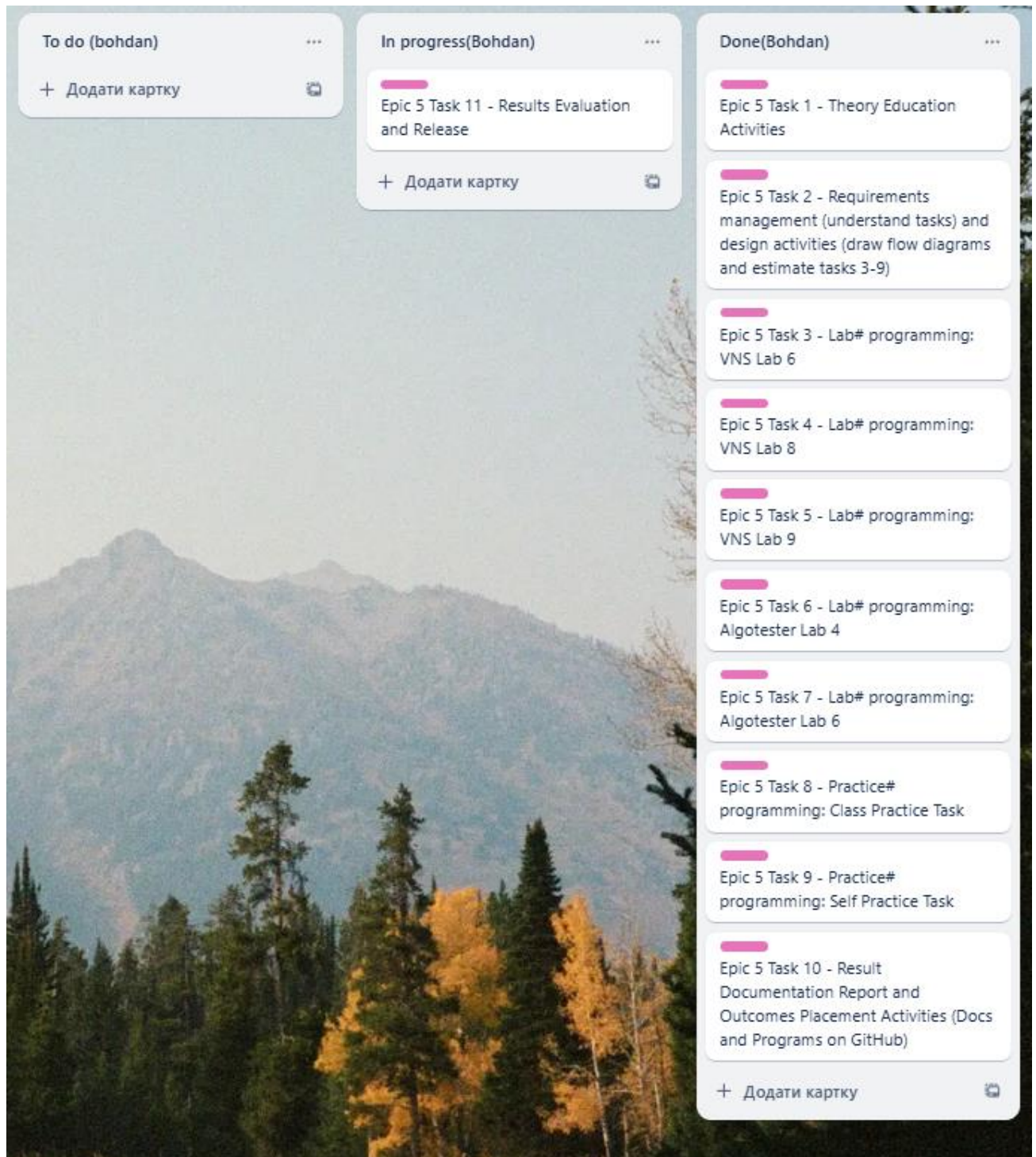
Зеник і Марічка приїхали до Антарктиди подивитися на пінгвінів. Мабуть, не знали, що у Львові вони теж є.

Усього є n пінгвінів, причому i -ий на початку розміщений у точці (x_i, y_i) на площині. Ціль пінгвінів — зустрітись разом у певній точці. На жаль, пінгвіни не дуже класно вміють ходити, тому за одну секунду з точки (x, y) пінгвін може перейти хіба що в $(x+1, y)$ або $(x, y+1)$. Ба більше, у будь-який момент часу рухатися може лише один пінгвін.

За який мінімальний час всі пінгвіни можуть зустрітись в певній точці?

1. Requirements management and design activities

Team Trello dashboard



Team meeting



UML-diagram block-scheme

VNS Lab 4 Task 1

35XB+-

VNS Lab 5 Task 1

20XB +-

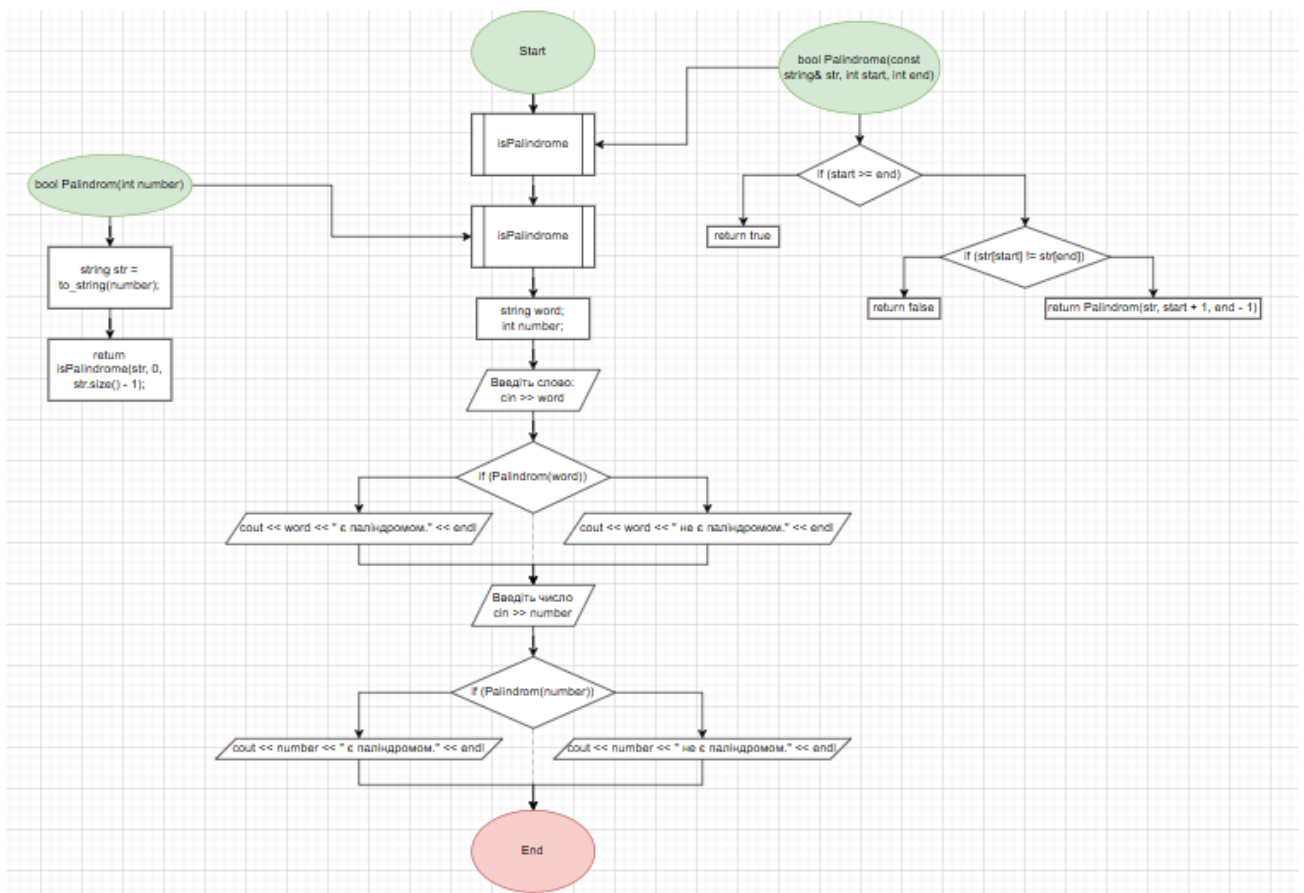
Algotester Lab 2 v3

35XB+-

ALgotester lab 3 v3

45XB+-

Class Practice Task



20XB+-

Self Practice Task

30XB+-

Код програми

VNS Lab 4 Task 1

```
#include <iostream>
#include <vector>
#include <ctime>
#include <cstdlib>

using namespace std;

void printArray(const vector<int>& arr) {
    for (int num : arr) {
        cout << num << " ";
    }
    cout << endl;
}

int main() {
    srand(static_cast<unsigned int>(time(0)));

    int n, indexToDelete;
    cout << "Введіть розмір масиву: ";
    cin >> n;

    vector<int> arr(n);

    for (int i = 0; i < n; i++) {
        arr[i] = rand() % 100;
    }

    cout << "Початковий масив: ";
    printArray(arr);

    cout << "Введіть індекс елемента для видалення (від 0 до " << n - 1 << "): ";
    cin >> indexToDelete;

    if (indexToDelete >= 0 && indexToDelete < arr.size()) {
        arr.erase(arr.begin() + indexToDelete);
    } else {
        cout << "Неправильний індекс!" << endl;
    }

    cout << "Масив після видалення елемента: ";
    printArray(arr);

    bool added = false;
    for (int i = 1; i < arr.size(); i++) {
        if (arr[i - 1] % 2 == 0) {
            int newElement = arr[i - 1] + 2;
            arr.insert(arr.begin() + i, newElement);
            added = true;
            break;
        }
    }

    if (!added) {
        cout << "Парного елемента не знайдено, додавання не виконане." << endl;
    }

    cout << "Масив після додавання елемента: ";
    printArray(arr);

    return 0;
}
```

```
Введіть розмір масиву: 3
Початковий масив: 84 56 41
Введіть індекс елемента для видалення (від 0 до 2): 2
Масив після видалення елемента: 84 56
Масив після додавання елемента: 84 86 56
PS E:\Projekts\c++\Epic 4>
```

+- 30 xB

VNS Lab 5 Task 1

```
#include <iostream>
#include <vector>
#include <limits>

using namespace std;

void findSaddlePoints(const vector<vector<int>>& matrix) {
    int rows = matrix.size();
    int cols = matrix[0].size();
    bool saddlePointFound = false;

    for (int i = 0; i < rows; i++) {
        int minInRow = matrix[i][0];
        int colIndex = 0;
        for (int j = 1; j < cols; j++) {
            if (matrix[i][j] < minInRow) {
                minInRow = matrix[i][j];
                colIndex = j;
            }
        }

        bool isSaddlePoint = true;
        for (int k = 0; k < rows; k++) {
            if (matrix[k][colIndex] > minInRow) {
                isSaddlePoint = false;
                break;
            }
        }

        if (isSaddlePoint) {
            cout << "Сідлова точка знайдена: " << minInRow << " на позиції (" << i << ", " << colIndex << ")" << endl;
            saddlePointFound = true;
        }
    }

    for (int i = 0; i < rows; i++) {
        int maxInRow = matrix[i][0];
        int colIndex = 0;
        for (int j = 1; j < cols; j++) {
            if (matrix[i][j] > maxInRow) {
                maxInRow = matrix[i][j];
                colIndex = j;
            }
        }

        bool isSaddlePoint = true;
        for (int k = 0; k < rows; k++) {
            if (matrix[k][colIndex] < maxInRow) {
                isSaddlePoint = false;
                break;
            }
        }

        if (isSaddlePoint) {
            cout << "Сідлова точка знайдена: " << maxInRow << " на позиції (" << i << ", " << colIndex << ")" << endl;
            saddlePointFound = true;
        }
    }

    if (!saddlePointFound) {
        cout << "Сідлових точок не знайдено." << endl;
    }
}

int main() {
    int rows, cols;
    cout << "Введіть кількість рядків і стовпців матриці: ";
    cin >> rows >> cols;

    vector<vector<int>> matrix(rows, vector<int>(cols));
    cout << "Введіть елементи матриці:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cin >> matrix[i][j];
        }
    }

    findSaddlePoints(matrix);

    return 0;
}
```

```
Введіть кількість рядків і стовпців матриці: 3 4
Введіть елементи матриці:
1 2 3 4
2 3 4 5
4 2 3 1
Сідлових точок не знайдено.
PS E:\Projekts\c++\Epic 4>
```

Algotester Lab 2 v3

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  pair<int, int> simulateDrones(const vector<int>& speeds, string& result) {
7      int left = 0;
8      int right = speeds.size() - 1;
9
10     while (true) {
11         if (left == right) {
12             result = "Collision";
13             return {left + 1, right + 1};
14         } else if (left + 1 == right) {
15             result = "Stopped";
16             return {left + 1, right + 1};
17         } else if (left > right) {
18             result = "Miss";
19             return {left + 1, right + 1};
20         }
21
22         left += speeds[left];
23         right -= speeds[right];
24     }
25 }
26
27 int main() {
28     int N;
29     cin >> N;
30     vector<int> speeds(N);
31
32     for (int i = 0; i < N; ++i) {
33         cin >> speeds[i];
34     }
35
36     string result;
37     auto finalPositions = simulateDrones(speeds, result);
38
39     cout << finalPositions.first << " " << finalPositions.second << endl;
40     cout << result << endl;
41
42     return 0;
43 }
44
```

Створено	Компилятор	Результат	Час (сек.)	Память (МБ)	Дл
4 години тому	C++ 23	Зараховано	0.003	1.230	Перегляд

+ - 20 xB


Algotester Lab 3 v3

```

1  ~ #include <iostream>
2  ~ #include <string>
3
4  ~ using namespace std;
5
6  ~ string compressString(const string& s) {
7      ~ string compressed;
8      ~ int count = 1;
9
10     ~ for (size_t i = 1; i <= s.size(); i++) {
11         ~ if (i < s.size() && s[i] == s[i - 1]) {
12             ~ count++;
13         ~ } else {
14             ~ compressed += s[i - 1];
15             ~ if (count > 1) {
16                 ~ compressed += to_string(count);
17             ~ }
18             ~ count = 1;
19         ~ }
20     ~ }
21
22     ~ return compressed;
23 ~ }
24
25 ~ int main() {
26     ~ string s;
27     ~ cin >> s;
28     ~ cout << compressString(s) << endl;
29     ~ return 0;
30 ~ }
31

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (MiB)	Дії
3 години тому	C++ 23	Зараховано	0.003	1.207	Перегляд

25 XB +-


Class Practice Task

```
#include <iostream>
#include <string>

using namespace std;

bool isPalindrome(const string& str, int start, int end) {
    if (start >= end) {
        return true;
    }
    if (str[start] != str[end]) {
        return false;
    }
    return isPalindrome(str, start + 1, end - 1);
}

bool isPalindrome(int number) {
    string str = to_string(number);
    return isPalindrome(str, 0, str.size() - 1);
}

int main() {
    string str;
    cout << "Введіть слово для перевірки: ";
    cin >> str;
    if (isPalindrome(str, 0, str.size() - 1)) {
        cout << "Рядок \"" << str << "\" є паліндромом." << endl;
    } else {
        cout << "Рядок \"" << str << "\" не є паліндромом." << endl;
    }

    int number;
    cout << "Введіть число для перевірки: ";
    cin >> number;
    if (isPalindrome(number)) {
        cout << "Число " << number << " є паліндромом." << endl;
    } else {
        cout << "Число " << number << " не є паліндромом." << endl;
    }

    return 0;
}
```

```
Введіть слово для перевірки: level
Рядок "level" є паліндромом.
Введіть число для перевірки: 101
Число 101 є паліндромом.
PS E:\Projekts\c++\Epic 4> |
```

+-- 40 xB

Self Practice Task

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main() {
8      int n;
9      cin >> n;
10
11     vector<pair<int, int>> penguins(n);
12
13     for (int i = 0; i < n; i++) {
14         cin >> penguins[i].first >> penguins[i].second;
15     }
16
17     int x_max = 0;
18     int y_max = 0;
19
20     for (int i = 0; i < n; i++) {
21         x_max = max(x_max, penguins[i].first);
22         y_max = max(y_max, penguins[i].second);
23     }
24
25     long long total_time = 0;
26
27     for (int i = 0; i < n; i++) {
28         total_time += (x_max - penguins[i].first) + (y_max - penguins[i].second);
29     }
30
31     cout << total_time << endl;
32
33     return 0;
34 }
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
2 години тому	C++ 23	Зараховано	0.092	2.098	Перегляд

+15 хв

Висновки:

Виконуючи цей епік я поглибив свої знання у зберіганні та обробці даних, що суттєво спрощує програмування. Застосування вказівників і динамічних масивів дозволяє зекономити пам'ять і гнучко управляти даними, а розуміння структур даних та алгоритмів обробки допомагає створювати оптимізовані рішення для різноманітних задач.

