МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ

**ЗВІТ**

**з лабораторної роботи №1**

**“Розробка та програмна реалізація словесних алгоритмів”**

Перевірив

Викладач: Лисицький В.Л

Виконала

студентка групи КН-222в

Григор’єва К.І.

Харків 2023

1. Тема

Розробка та програмна реалізація словесних алгоритмів.

2. Мета

Надбання навичок словесного опису алгоритмів розв’язання реальних задач.

3. Теоретичні основи

T1 – предметна технологія визначення розмахів вибіркової сукупності спостережень (вибірка) протягом часу з використанням програмного забезпечення (ПЗ) складної системи. Розмах вибірки – різниця між максимальним та мінімальних елементами вибірки.

Відповідь на наступни запитання:

1. Навіщо потрібна математична статистика для програмного проєктування?

Математична статистика є важливим інструментом для програмного проєктування з кількох причин. По-перше, вона допомагає зрозуміти дані, що збираються в програмах, та знайти закономірності, які можуть бути корисними для вдосконалення функцій програми. Наприклад, статистичний аналіз може допомогти визначити, які дані є найбільш важливими для певної функції програми, що допоможе її оптимізувати.

По-друге, математична статистика допомагає оцінити ефективність програмного забезпечення, наприклад, оцінити, які функції потребують більше часу для виконання, які функції є найбільш ресурсозатратними, та які алгоритми є найефективнішими для виконання певних завдань.

По-третє, математична статистика може бути корисною для розв'язання проблем, пов'язаних з обробкою даних у програмах. Наприклад, аналіз даних може допомогти програмістам виявити неправильну поведінку програми або помилки в коді, що важливо для забезпечення безпеки програми.

2. Чи потрібна теорія алгоритмів для проєктування програмного забезпечення?

Теорія алгоритмів також є важливим інструментом для програмного проєктування, оскільки вона допомагає розробити оптимальні алгоритми для розв'язання певних завдань. Це особливо важливо для програм, які вимагають обробки великої кількості даних, оскільки ефективність алгоритму може суттєво вплинути на швидкість виконання програми.

Також теорія алгоритмів може бути корисною для програмістів при розробці програмного забезпечення, що потребує обробки складних завдань, наприклад, в галузі штучного інтелекту, обробки природних мов або комп'ютерного зору. Знання теорії алгоритмів допоможе програмістам розробити оптимальні алгоритми, що відповідають вимогам ефективності та швидкодії програмного забезпечення.

**Завдання до лабораторної роботи:**

**1.** Визначимо номер K варіанту студента для визначення елементів матриці Аij, з формулою: *K = j + r*, де j - номер прізвища студента у списку групи, a r – номер групи, тобто r = 3(група В). У моєму варіантів К = 12 + 3 = 15.

**2.** Обчислимо значення елементів Аij матриці А за допомогою формули: *A=K \*Ặ*, де К – номер студента в списку групи, у моєму випадку це №12.

**3.1.** Розробка словесного алгоритму:

1. Визначення розмаху RA вибірки, матрицею: А = (aij), i = 1, m, j = 1, n. RA = Аmax - Amin, де Аmax = aij (max), Amin = aij (min).
2. Визначення розмаху кожного стовпця: Rj = (aij), i = 1, n матриці А; розмаху

RAj, j = 1, n, RAj = Aj max - Aj min , де Aj max = max (aij) i = 1, m, Aj min = min (aij) i = 1, m.

1. Кожного рядка Ai = (aij), j = 1, n матриці А; розмаху RA i , i = 1, m, RA i = A i max – A i min, де A i max = max (aij) j = 1, n, Ai min = min (aij) j = 1, n.
2. Мінімальний RAmin розмах стовпців Aj вибірки А, максимальний RAmax,

RA max = max (RAj) j = 1, n, RAmin = min (RAj) j = 1, n.

1. Мінімальний RAmin, RAmax розмах рядків вибірки А,

Rmax = max (RAi) i = 1, m, Rmin = min (RAi) i = 1, m.

1. Результатом μ словесного алгоритму є кортеж μ = {RA, RAmax , RA min , Rmax, Rmin}

***Словесний алгоритм:***

**|20,3 15,4 17,2 19,2 23,3 18,1 21,9|**

**|15,3 16,8 13,2 20,4 16,5 19,7 20,5|**

**|14,3 20,1 16,8 14,7 20,8 19,5 15,3|**

**|19,3 17,8 16,2 15,7 22,8 21,9 12,5|**

**Ᾱ = |10,1 21,1 18,3 14,7 14,5 18,1 18,4|**

**|13,9 19,1 18,5 20,2 23,8 16,7 20,4|**

**|19,5 17,2 19,6 17,8 21,3 17,5 19,4|**

**|17,8 13,5 17,8 11,8 18,6 19,1 19,5|**

**Рис.1**

**Словесний алгоритм**

1. Вибрати значення k, за яким будуть обчислюватись значення матриці А. Перейти до п.2.
2. Наявну матрицю Ᾱ (рис.1), що має встановлену кількість рядків та стовпців, де i = 8 та j = 7 відповідно. Перейти до п.3.
3. Обрахувати матрицю А, шляхом множення кожного елементу матриці a­ij матриці А на значення k(з п.1). Перейти до п 4.1:
   1. . Для початкового значення обираємо елемент a00, множимо його значення k(з п.1). Усі інші елементи матриці А також множити на множник k.
   2. . Встановити значення циклічної змінної i = 0. Перейти до п. 3.3.
   3. . Якщо i => 8, негайно перейти до п. 4. Якщо ні, перейти до п. 3.4.
   4. . Встановити значення циклічної змінної j = 0. Перейти до п. 3.5.
   5. . Якщо j => 7, негайно перейти до п. 4. Якщо ні, продовжити множення. Перейти до п. 3.6.
   6. . Збільшити значення j на 1 (j = j + 1). Перейди до п. 3.5.
   7. . Збільшити значення i на 1 (i = i + 1). Перейти до п. 3.3.
4. Знаходимо максимальне значення елементів Аmax та Аmin з отриманої матриці А (п.3). Перейти до п. 4.1:
   1. . Встановити значення min\_matrix = matrix[0][0] та max\_matrix = matrix[0][0]. Перейти до п. 4.2.
   2. . Встановити значення циклічної змінної i = 0. Перейти до п. 4.3.
   3. . Якщо i => 8, негайно перейти до п. 5. Якщо ні, перейти до п. 4.3.1.
      1. Встановити значення циклічної змінної j = 0. Перейти до п. 4.3.2.
      2. Якщо j => 7, негайно перейти до п. 4.4. Якщо ні, перейти до п. 4.3.3.
      3. Порівняти matrix[i][j] з min\_matrix. Якщо matrix[i][j] < min\_matrix, то присвоїти min\_matrix = matrix[i][j]. Перейти до п. 4.3.4.
      4. Порівняти matrix[i][j] з max\_matrix. Якщо matrix[i][j] > max\_matrix, то присвоїти max\_matrix = matrix[i][j]. Перейти до п. 4.3.5.
      5. Збільшити значення j на 1 (j = j + 1). Перейди до п. 4.3.2.
   4. . Збільшити значення i на 1 (i = i + 1). Перейти до п. 4.3.
5. Обчислити розмах RA матриці A. RA = Аmax - Amin. Перейти до п.6
6. Знаходимо максимальне та мінімальне значення елементів кожного стовпчика матриці A: Перейти до п. 6.1:
   1. . Встановити значення циклічної змінної j = 0. Встановити значення циклічної змінної j = 0. Перейти до п. 6.2.
   2. . Якщо j >= 7, негайно перейти до п. 7. Якщо ні, перейти до п. 6.3.
   3. . Встановити значення змінних min\_col та max\_col. Перейти в п. 6.4
   4. . Встановити значення циклічної змінної i = 0. Перейти до п. 6.5.
   5. . Якщо i >= 8, негайно перейти до п. 7. Якщо ні, перейти до п. 6.6.
   6. . Порівняти значення елемента matrix[i][j] зі значенням змінних min\_col та max\_col. Якщо matrix[i][j] < min\_col, присвоїти min\_col значення matrix[i][j]. Якщо matrix[i][j] > max\_col, присвоїти max\_col значення matrix[i][j]. Перейти до п. 6.7.
   7. . Знайти розмах кожного з стовпців [j] за формулою max\_col - min\_col і зберегти результат в п. 7. Перейти до п. 6.8.
   8. . Збільшити значення i на 1 (i = i + 1). Перейти до п. 6.5.
   9. . Збільшити значення j на 1 (j = j + 1). Перейти до п. 6.2.
7. Зберегти результати п. 6.7. для подальшої обробки у п. 10. Перейти до п. 8.
8. Знаходимо максимальне та мінімальне значення елементів кожного рядка матриці A: Перейти до п. 8.1:
   1. . Встановити значення циклічної змінної j = 0. Перейти до п. 8.2.
   2. . Якщо i >= 8, негайно перейти до п. 9. Якщо ні, перейти до п. 8.3.
   3. . Встановити значення змінних min\_row та max\_row. Перейти в п. 8.4
   4. . Встановити значення циклічної змінної i = 0. Перейти до п. 8.5.
   5. . Якщо j >= 7, негайно перейти до п. 9. Якщо ні, перейти до п. 8.6.
   6. . Порівняти значення елемента matrix[i][j] зі значенням змінних min\_row та max\_row. Якщо matrix[i][j] < min\_row, присвоїти min\_row значення matrix[i][j]. Якщо matrix[i][j] > max\_row, присвоїти max\_row значення matrix[i][j]. Перейти до п. 8.7.
   7. . Знайти розмах кожного з рядків [i] за формулою max\_row - min\_row і зберегти результат в п. 9. Перейти до п. 8.
   8. . Збільшити значення j на 1 (i = i + 1). Перейти до п. 8.5.
   9. . Збільшити значення i на 1 (j = j + 1). Перейти до п. 8.2.
9. Зберегти результати п. 8.7. для подальшої обробки у п.13. Перейти у п. 10
10. Знаходимо найбільше(RAmax) і найменше(RAmin) значення розмаху серед стовпців у матриці А: Перейти до п. 10.1:

***10.1.*** Отримати вхідні дані з п. 7 .Встановити початкові значення для min\_scope\_col та max\_scope\_col як найбільше та найменше значення в стовпці відповідно. Перейти до п. 10.2.

***10.2.*** Встановити значення циклічної змінної j = 0. Перейти до п. 10.3.

***10.3.*** Якщо j >= 7, негайно перейти до п. 11. Якщо ні, перейти до п. 10.4.

***10.4.*** Якщо cols\_scope < min\_scope\_col, встановити значення min\_scope\_col = cols\_scope. Перейти до п. 10.5.

**10.5.** Якщо cols\_scope > max\_scope\_col, встановити значення max\_scope\_col = cols\_scope. Перейти до п. 10.6.

**10.6.** Збільшити значення j на 1 (j = j + 1). Перейти до п. 3.3.

1. Записати результат роботи п. 10. Перейти до п. 12.
2. Знаходимо найбільше(Rmax) і найменше(Rmin) значення розмаху серед рядків у матриці А: Перейти до п. 11.1:

***11.1.*** Отримати вхідні дані з п. 9. Встановити початкові значення для min\_scope\_row та max\_scope\_row як найбільше та найменше значення в стовпці відповідно. Перейти до п. 10.2

***11.2.*** Встановити значення циклічної змінної i = 0. Перейти до п. 11.3.

***11.3.*** Якщо i >= 8, негайно перейти до п. 13. Якщо ні, перейти до п. 11.4.

***11.4.*** Якщо rows\_scope < min\_scope\_row, встановити значення min\_scope\_row = rows\_scope. Перейти до п. 11.5.

**11.5.** Якщо rows\_scope > max\_scope\_row, встановити значення max\_scope\_row = rows\_scope. Перейти до п. 11.6.

**11.6.** Збільшити значення i на 1 (i = i + 1). Перейти до п. 11.3.

1. Записати результат роботи п.12. Перейти до п. 14.
2. Записати кінцевий результат словесного алгоритму у вигляді КОРТЕЖУ U = {RA, RAmax, RAmin, Rmax, Rmin}. ***Алгоритм завершено.***

**3.2. Розробка тесту для тестування програмного рішення задачі. Для вибіркової сукупності спостережень (вибірки), яка визначається матрицею А = k \* A (див. п. 2.4) першим способом відповідно словесному алгоритму отримати матрицю.**

1. Була взята матриця

{20.3, 15.4, 17.2, 19.2, 23.3, 18.1, 21.9},

{15.3, 16.8, 13.2, 20.4, 16.5, 19.7, 20.5},

{14.3, 20.1, 16.8, 14.7, 20.8, 19.5, 15.3},

{19.3, 17.8, 16.2, 15.7, 22.8, 21.9, 12.5},

{10.1, 21.1, 18.3, 14.7, 14.5, 18.1, 18.4},

{13.9, 19.1, 18.5, 20.2, 23.8, 16.7, 20.4},

{19.5, 17.2, 19.6, 17.8, 21.3, 17.5, 19.4},

{17.8, 13.5, 17.8, 11.8, 18.6, 19.1, 19.5}

1. Підставлення своїх даних j,r.
2. Обчислювання даних k = j + r
3. Обчислення матриці за допомогою формули *A=K \*Ặ*
4. Визначаємо розмах RA вибірки, матрицею: А = (aij), i = 1, m, j = 1, n. RA = Аmax - Amin, де Аmax = aij (max), Amin = aij (min).
5. Визначаємо розмах кожного стовпця: Rj = (aij), i = 1, n матриці А; розмаху RAj, j = 1, n, RAj = Aj max - Aj min , де Aj max = max (aij) i = 1, m, Aj min = min (aij) i = 1, m.
6. Визначаємо розмах кожного рядка Ai = (aij), j = 1, n матриці А; розмаху RA i , i = 1, m, RA i = A i max – A i min, де A i max = max (aij) j = 1, n, Ai min = min (aij) j = 1, n.
7. Визначаємо Мінімальний RAmin розмах стовпців Aj вибірки А, максимальний RAmax, RA max = max (RAj) j = 1, n, RAmin = min (RAj) j = 1, n.
8. Визначаємо Мінімальний RAmin, RAmax розмах рядків вибірки А, Rmax = max (RAi) i = 1, m, Rmin = min (RAi) i = 1, m.
9. Отримання результату словесного алгоритму та перевірка його з ручним підрахунком.

**3.3 Написання тексту програми**

Код програми

#include <iostream>

#include <tuple>

using namespace std;

const int rows = 8;

const int cols = 7;

void funcmatrix(double matrix[rows][cols], const int rows, const int cols, int k)

{

for (int indexCols = 0; indexCols < cols; indexCols++)

{

for (int indexRows = 0; indexRows < rows; indexRows++)

{

matrix[indexRows][indexCols] = k \* matrix[indexRows][indexCols];

}

}

}

double funcRA(double matrix[rows][cols], const int rows, const int cols)

{

double Amax = matrix[0][0];

double max = matrix[0][0];

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

if (matrix[i][j] > max)

{

max = matrix[i][j];

}

}

if (max > Amax)

{

Amax = max;

}

}

double Amin = matrix[0][0];

double min = matrix[0][0];

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

if (matrix[i][j] < min)

{

min = matrix[i][j];

}

}

if (min < Amin)

{

Amin = min;

}

}

return Amax - Amin;

}

void RAmin\_AND\_RAmax(double matrix[rows][cols], double \*RAmax, double \*RAmin, const int rows, const int cols)

{

double RAJ = 0;

for (int j = 0; j < cols; j++)

{

double maxAJ = matrix[0][j];

double minAJ = matrix[0][j];

for (int i = 0; i < rows; i++)

{

if (matrix[i][j] < minAJ)

{

minAJ = matrix[i][j];

}

if (matrix[i][j] > maxAJ)

{

maxAJ = matrix[i][j];

}

}

if (RAJ == 0)

{

RAJ = maxAJ - minAJ;

\*RAmax = RAJ;

\*RAmin = RAJ;

}

else

{

RAJ = maxAJ - minAJ;

if (\*RAmin > RAJ)

{

\*RAmin = RAJ;

}

else if (\*RAmax < RAJ)

{

\*RAmax = RAJ;

}

}

}

}

void Rmin\_AND\_Rmax(double matrix[rows][cols], double \*Rmax, double \*Rmin, const int rows, const int cols)

{

double RAI = 0;

for (int i = 0; i < rows; i++)

{

double maxAI = matrix[i][0];

double minAI = matrix[i][0];

for (int j = 0; j < cols; j++)

{

if (matrix[i][j] < minAI)

{

minAI = matrix[i][j];

}

if (matrix[i][j] > maxAI)

{

maxAI = matrix[i][j];

}

}

if (RAI == 0)

{

RAI = maxAI - minAI;

\*Rmax = RAI;

\*Rmin = RAI;

}

else

{

RAI = maxAI - minAI;

if (\*Rmin > RAI)

{

\*Rmin = RAI;

}

else if (\*Rmax < RAI)

{

\*Rmax = RAI;

}

}

}

}

int main()

{

int j = 8;

int r = 3;

cout << "Enter your number behind the group list: " << endl;

cin >> j;

cout << "Enter your group number(A - 1, B - 2, C - 3, D - 4): " << endl;

cin >> r;

int k = j + r;

double matrix[rows][cols] = {{20.3, 15.4, 17.2, 19.2, 23.3, 18.1, 21.3},

{15.3, 16.8, 13.2, 20.4, 16.5, 19.7, 20.5},

{14.3, 20.1, 16.8, 14.7, 20.8, 19.5, 15.3},

{19.3, 17.8, 16.2, 15.7, 22.8, 21.9, 12.5},

{10.1, 21.1, 18.3, 14.7, 14.5, 18.1, 18.4},

{13.9, 19.1, 18.5, 20.2, 23.8, 16.7, 20.4},

{19.5, 17.2, 19.6, 17.8, 21.3, 17.5, 19.4},

{17.8, 13.5, 17.8, 11.8, 18.6, 19.1, 19.5}};

// Matrix

funcmatrix(matrix, rows, cols, k);

// RA

double RA = funcRA(matrix, rows, cols);

// RAmax, RAmin

double RAmax = 0;

double RAmin = 0;

RAmin\_AND\_RAmax(matrix, &RAmax, &RAmin, rows, cols);

// Rmax, Rmin

double Rmax = 0;

double Rmin = 0;

Rmin\_AND\_Rmax(matrix, &Rmax, &Rmin, rows, cols);

auto progRes = make\_tuple(RA,RAmax,RAmin,Rmax,Rmin);

cout << "The values of tuple are: ";

cout << "\nRA is: " << get<0>(progRes) << "\nRAmax is: " << get<1>(progRes) << "\nRAmin is: " << get<2>(progRes) << "\nRmax is: " << get<3>(progRes) << "\nRmin is: " << get<4>(progRes)<< endl;

return 0;

}

**3.4. Опис представлення роботи з програмою.**

1. Увімкнення комп'ютера
2. Запуск програми Visual Studio Code
3. Відкриття термінала у Visual Studio Code
4. Компіляція в терміналі за допомогою команди "g++ НазваФайлу.cpp -o НазваФайлу.o"
5. Запуск програми в терміналі за допомогою команди "./НазваФайлу.o"
6. Отримання результату
7. Закриття програми Visual Studio Code
8. Вимкнення комп'ютера

Висновок: створений алгоритм у програмному середовища Microsoft Visual Studio при введені заданих параметрів працює коректно та швидко.

**3.5. Тестування програмного рішення задачі, а також чисельні експерименти**

1. Взята наступна матриця

{20.3, 15.4, 17.2, 19.2, 23.3, 18.1, 21.9},

{15.3, 16.8, 13.2, 20.4, 16.5, 19.7, 20.5},

{14.3, 20.1, 16.8, 14.7, 20.8, 19.5, 15.3},

{19.3, 17.8, 16.2, 15.7, 22.8, 21.9, 12.5},

{10.1, 21.1, 18.3, 14.7, 14.5, 18.1, 18.4},

{13.9, 19.1, 18.5, 20.2, 23.8, 16.7, 20.4},

{19.5, 17.2, 19.6, 17.8, 21.3, 17.5, 19.4},

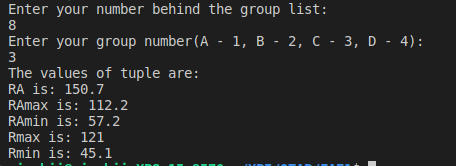
{17.8, 13.5, 17.8, 11.8, 18.6, 19.1, 19.5}

1. Підставлені наступни значення

j = 8

r = 3

1. Отримано наступний результат у консолі



1. Перевірка результату із консолі з ручним підрахунком (успішний результат)

**Висновок:** під виконання цієї лабораторної роботи я отримала навички словесного опису алгоритмів розв’язання реальних задач, опису тесту для тестування ПЗ і самого тестування ПЗ, навчилася описувати рішення програмних задач.

**Джерела використаної інформації:**

* **Горлова Т.М.**Теорія алгоритмів. [Електронний ресурс]: конспект лекцій для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навчання / Т.М. Горлова, К.Є. Бобрівник, Н.В. Ліманська – К.: НУХТ, 2015. – 95 с.
* https://elearning.sumdu.edu.ua/free\_content/lectured:5de5178bb62ca7a97fe35cba8b92d1b337ee8101/latest/8105/index.htmlHgfhgghы