МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІНСТИТУТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ФУНДАМЕНТАЛЬНИХ НАУК

Кафедра прикладної математики

**ЗВІТ**

про виконання лабораторної роботи №1 із дисципліни:

“**Математичні основи штучного інтелекту”**

Виконав:

студент групи ПМ-31

Матолін Марко

Прийняв:

доцент кафедри ПМ

Пабирівський В.В.

Львів ‒ 2023

**Тема:** «Алгоритм відпалу»

# Завдання

Розв’язати із використанням алгоритму відпалу задачу розстановки N шахових ферзів на шаховій дошці розміру N×N таким чином, аби жоден ферзь не загрожував будьякому іншому.

# Етапи виконання завдання

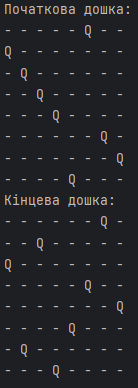
1. Вивчити із використанням запропонованих літературних джерел зміст алгоритму відпалу.
2. Обрати для визначеності фіксоване значення N (для прикладу, стандартний розмір шахової дошки 8×8, тому можна покласти N=8).
3. Реалізація допоміжних функцій випадкової зміни розв’язку та початкової ініціалізації.
4. Реалізація допоміжної функції для оцінки розв’язку.
5. Реалізація допоміжної функції копіювання одного розв’язку в інший.
6. Реалізація допоміжної функції виводу результату на екран у вигляді шахової дошки.
7. Безпосередня реалізація алгоритму відпалу.
8. Знайти інші алгоритми розв’язку задачі та спробувати розробити програмний код для їх реалізації.

# Варіант-3

Код програми:

import random  
import math  
  
def initial\_state(n):  
 state = list(range(n))  
 random.shuffle(state)  
 return state  
  
def conflicts(state):  
 n = len(state)  
 count = 0  
 for i in range(n):  
 for j in range(i+1, n):  
 if state[i] == state[j] or abs(state[i]-state[j]) == abs(i-j):  
 count += 1  
 return count  
  
def anneal(n, initial\_temp, cooling\_rate, max\_iterations):  
 current\_state = initial\_state(n)  
 current\_energy = conflicts(current\_state)  
 best\_state = current\_state.copy()  
 best\_energy = current\_energy  
 temperature = initial\_temp  
  
 for i in range(max\_iterations):  
 if current\_energy == 0:  
 break  
  
 new\_state = current\_state.copy()  
 a, b = random.sample(range(n), 2)  
 new\_state[a], new\_state[b] = new\_state[b], new\_state[a]  
 new\_energy = conflicts(new\_state)  
  
 if new\_energy == 0:  
 current\_state = new\_state  
 current\_energy = new\_energy  
 best\_state = new\_state  
 best\_energy = new\_energy  
 else:  
 delta\_energy = new\_energy - current\_energy  
 if delta\_energy < 0:  
 current\_state = new\_state  
 current\_energy = new\_energy  
 else:  
 p = math.exp(-delta\_energy / temperature)  
 if random.random() < p:  
 current\_state = new\_state  
 current\_energy = new\_energy  
  
 temperature \*= cooling\_rate  
  
 return best\_state  
  
n = 8  
initial\_temp = 100  
cooling\_rate = 0.95  
max\_iterations = 10000  
  
initial\_board = initial\_state(n)  
  
print('Початкова дошка:')  
for row in range(n):  
 line = ''  
 for col in range(n):  
 if initial\_board[row] == col:  
 line += 'Q '  
 else:  
 line += '- '  
 print(line)  
  
solution = anneal(n, initial\_temp, cooling\_rate, max\_iterations)  
  
print('Кінцева дошка:')  
for row in range(n):  
 line = ''  
 for col in range(n):  
 if solution[row] == col:  
 line += 'Q '  
 else:  
 line += '- '  
 print(line)

**Результат виконання:**



**Висновок:**

На лабораторній роботі я ознайомилась та застосувала ра практиці алгоритм відпалу