слугує чудовим прикладом для ознайомлення з основами криптографії та розуміння принципів роботи шифрів. Ця реалізація може бути використана як основа для подальшого вивчення та розробки більш складних криптографічних алгоритмів.

Список використаних джерел:

1. Fastovets V. Аналіз та програмна реалізація модифікованого криптографічного шифру Вернама та шифру Цезаря. Vehicle and electronics. Innovative technologies, 2021, (20), 62-67.
2. Bhanot R., Hans R. A review and comparative analysis of various encryption algorithms. 2015. Int. J. Secur. its Appl. 9. 4. 289–306.

Annotation: *The Caesar cipher is one of the oldest and simplest cryptographic algorithms used to protect confidential information. Despite its simplicity, this method of encryption is still widely used to introduce the basics of cryptography and understand how ciphers work. In this talk, we will look at the implementation of the Caesar cipher algorithm in the Python programming language, and we will also create a graphical user interface (GUI) using the tkinter module.*
Key words: *Caesar cipher, encryption, Python, tkinter module.*

Науковий керівник: Тищенко С.І.,
к.п.н., доцент,
доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна

УДК 004.41

РЕАЛІЗАЦІЯ ПОПУЛЯРНИХ АЛГОРИТМІВ СОРТУВАННЯ МОВОЮ PYTHON

Вишневський Олег Олександрович,
здобувач вищої освіти спеціальності 122 « Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

Анотація: *В дослідженні проаналізовано найпростіші алгоритми сортування та наведено програмний код їх реалізації в мові програмування Python. Розглянуто алгоритми сортування бульбашкою або Bubble Sort, сортування за вибором або Selection Sort та сортування вставками або Insertion Sort.*

Ключові слова: *python, алгоритми сортування, сортування бульбашкою, сортування за вибором, сортування вставками.*

Сортування бульбашкою або **Bubble Sort** – вид сортування, що має одну з максимально зрозумілих концепцій сортування. Основна ідея бульбашкового сортування полягає в порівнянні сусідніх елементів масиву та їх обміні, якщо вони знаходяться в неправильному порядку. Сортування продовжується до тих пір, поки всі елементи не будуть відсортовані.

Основні кроки бульбашкового сортування:

1. Порівнюємо два сусідні елементи масиву.
2. Якщо вони знаходяться в неправильному порядку, то міняємо їх місцями.
3. Повторюємо ці кроки для всього масиву, здійснюючи ітерації, поки масив не буде відсортований повністю.

Реалізація алгоритму сортування бульбашкою мовою Python:

```
def bubble_sort(array):
    arrLength = len(array)
    for i in range(0, arrLength):
        for j in range(0, arrLength - i - 1):
            if array[j] > array[j + 1]:
                temp = array[j]
                array[j] = array[j + 1]
                array[j + 1] = temp
print("Реалізація алгоритму сортування бульбашкою")
arr = []
n = int(input("Введіть кількість елементів масиву: "))
for i in range(0, n):
    elem = int(input("Елемент " + str(i + 1) + " = "))
    arr.append(elem)
bubble_sort(arr)
print("Відсортований масив:")
print(arr)
```

Сортування за вибором або **Selection Sort** – вид сортування, що більш ефективний, ніж бульбашковий. Він відноситься до категорії алгоритмів сортування вибору.

Основні кроки сортування за вибором:

1. Знаходимо мінімальний елемент.
2. Обмінюємо мінімальний елемент з першим елементом масиву.
3. Збільшуємо область впорядкування.
4. Повторюємо дії до кінця сортування масиву.

Реалізація алгоритму сортування за вибором мовою Python:

```
def sel_sort(array):
    for i in range(len(array)):
        min_index = i
        for j in range(i + 1, len(array)):
```

```
            if array[j] < array[min_index]:
                min_index = j
        array[i], array[min_index] = array[min_index], array[i]
    return(array)
print("Реалізація алгоритму сортування за вибором")
arr = []
n = int(input("Введіть кількість елементів масиву: "))
for i in range(0, n):
    elem = int(input("Елемент " + str(i + 1) + " = "))
    arr.append(elem)
sel_sort(arr)
print("Відсортований масив:")
print(arr)
```

Сортування вставками або **Insertion Sort** – є ще одним простим алгоритмом сортування. Його основна ідея полягає в тому, щоб поступово будувати впорядковану послідовність елементів, вставляючи кожен на відповідне місце. Сортування вставками є ефективним для невеликих обсягів даних або для вже частково впорядкованих масивів, оскільки в обох випадках воно може працювати досить швидко.

Основні кроки сортування вставками:

1. Припускаємо, що перший елемент масиву вже вважається впорядкованим.
2. Вставляємо елементи, де з кожним новим елементом алгоритм його порівнює з вже впорядкованою частиною масиву та вставляє на відповідне йому місце, згідно впорядкованості.
3. Збільшуємо впорядковану область.
4. Повторюємо кроки під номером 2 і 3 для всіх елементів масиву, доки масив не буде весь впорядкований.

Реалізація алгоритму сортування вставками мовою Python:

```
def insert_sort(array):
    for i in range(1,len(array)):
        key = array[i]
        p = i
        while p - 1 >= 0 and key < array[p-1]:
            array[p - 1], array[p] = array[p], array[p - 1]
            p -= 1
        array[p] = key
    return(array)
print("Реалізація алгоритму сортування вставками")
arr = []
n = int(input("Введіть кількість елементів масиву: "))
for i in range(0, n):
    elem = int(input("Елемент " + str(i + 1) + " = "))
    arr.append(elem)
insert_sort(arr)
```

```
print("Відсортований масив:")
print(arr)
```

Результат роботи всіх трьох розглянутих алгоритмів сортування однаковий (скріншот з сервісу replit.com):

```
Реалізація алгоритму сортування
Введіть кількість елементів масиву: 5
Елемент 1 = 7
Елемент 2 = 2
Елемент 3 = 8
Елемент 4 = 1
Елемент 5 = 3
Відсортований масив:
[1, 2, 3, 7, 8]
```

Список використаних джерел:

1. https://teletype.in/@cursor.education
2. https://sites.google.com/view/python-11-profilniy/
3. https://www.miyklas.com.ua/p/informatica/9-klas/algoritmi-i-programi-python-447430/
4. https://www.youtube.com/watch?v=5MqqAIalkNw

Annotation: *The research analyzes the simplest sorting algorithms and provides the program code for their implementation in the Python programming language. Algorithms of Bubble Sort, Selection Sort and Insertion Sort are considered.*

Key words: *Python, sorting algorithms, bubble sort, selection sort, insertion* sort.

Науковий керівник: Пархоменко О.Ю.,
к.ф.-м.н., доцент,
доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна

УДК 004.222(07)

ДВІЙКОВА ТА ІНШІ СИСТЕМИ ЧИСЛЕННЯ

Власов Артем Сергійович,
здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

Анотація: *Досліджено основні концепції та принципи числових систем, які є важливими в інформатиці та математиці. Розглянуто особливості двійкової системи, що базується на використанні всього двох символів 0 і 1, що робить її надзвичайно важливою для комп'ютерних систем та цифрових технологій. Розглядаються також інші системи числення, такі як*