

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М.Є. Жуковського  
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки (503)

Лабораторна робота № 5

Дослідження організації даних методом розстановки

(назва лабораторної роботи)

з дисципліни

Моделі та структури даних

(шифр)

ХАІ.503.525а.03О.123-Комп'ютерна інженерія, ПЗ №9629619

Виконав студент гр.  
22.11.2022

525а  
(№ групи)

Литвиненко А.В.  
(П.І.Б.)

\_\_\_\_\_  
(підпис, дата)

Перевірив

канд. техн. наук, доцент

\_\_\_\_\_  
(підпис, дата)

А. В. Шостак  
(П.І.Б.)

Харків – 2022

**Тема роботи:** дослідження організації даних методом розстановки.

## Варіант 5

### Задача 1

#### Частина 1. Постановка завдання

##### Умова:

1. Изучить теоретический материал по теме лабораторной работы в размере конспекта лекций и данного пособия и ответить на контрольные вопросы.

2. Заполнить разделы отчета по лабораторной работе.

3. Из таблицы 1 выбрать  $n=8$  чисел в соответствии с формулой (1)

$$N1 = S + K(S \bmod 2 + 1), \quad (1)$$

где  $N1$  – номер первого ключа последовательности,  $S$  – номер по списку,  $K$  – порядковый номер метода открытой адресации для разрешения коллизий в соответствии с таблицей 2.

4. Создать таблицу расстановки последовательности ключей из пункта 3 с помощью методов открытой адресации в соответствии с таблицей 2 и хеш-функцией  $H(k) = k \bmod N, N=13$ .

5. Написать программу, которая создаёт таблицу расстановки последовательности ключей с помощью хеш-функции  $H(k) = k \bmod N$  и разрешает коллизии с помощью заданных в таблице 2 методов открытой адресации.

##### При выполнении лабораторной работы

1. Отладить программу исследования данных методом расстановки.

2. Определить для исследуемых методов открытой адресации из таблицы 2 среднее число обращений к таблице расстановки при вставке и среднее число обращений при результативном поиске как функцию от коэффициента заполнения таблицы расстановки  $\alpha = n/N$ , ( $\alpha = \{0.25, 0.50, 0.75, 0.90, 0.95\}$ ).

3. В качестве последовательности ключей использовать целые числа, равномерно распределенные в интервале  $[A, B]$ . Для формирования последовательности ключей использовать датчик равномерно распределенных чисел  $rand()$ .

4. На основе проведенных исследований обобщить полученные результаты, сделать выводы по работе, оформить отчет по лабораторной работе и представить его преподавателю на защиту.

##### Умова з додатка:

5	23	20	90	25	60	99	01	90	25	29
---	----	----	----	----	----	----	----	----	----	----

5.	[0,1000]	83	2, 3, 4
----	----------	----	---------

Варианты методов открытой адресации:

- 2) метод линейных проб ( $H_i = (H_{i-1} + c) \bmod N$ );
- 3) метод квадратичных проб;
- 4) метод двойного хеширования;

## Частина 2. Схема класу

На основі постановки завдання розроблена схема, представлений на рисунку 1.

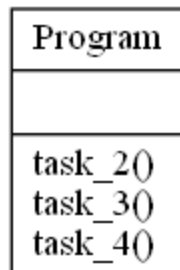


Рисунок 1 - Схема класу перетворення

#### Частина 4. Текст програми

Відповідно до розробленого алгоритму в середовищі Microsoft Visual Studio була написана програма, яка наведена нижче.

##### Main.py

```
import math
from pprint import pprint
# File: main.py
# Author: Lytvynenko Andrii 525a
# Date: 27.11.2022

# Інтерва розподілу чисел [A,B]
Interval = [0,1000]
# Розмірність таблиці
N = 83

# Рівномірно розподілені числа
Table1_Array = [23, 20, 90, 25, 60, 99, 1, 90, 25, 29]

# Для методу лінійних проб
N1 = 5 + 2*(5%2+1) # 9
# Для квадратичних проб
N2 = 5 + 3*(5%2+1) # 11
# Метод подвійного хешування
N3 = 5 + 4*(5%2+1) # 13
L = N3 - 3

# N1 = S+K(S % 2 + 1)
# N1 - номер першого ключа послдовательности
# S - номер по списку
# K - порядковий номер метода відкритої адресації для дозволу колізії в табл 2

# Варіанти методів відкритої адресації
# 2) метод лінійних проб ( $H.i = (H.i-1 + c) \% N$ )
# 3) метод квадратичних проб
# 4) метод подвійного хешування

"""
Значення з презентації
Table1_Array = [1,0,13,26,39]
N = 13
c = 2
"""
```

```

def H(k: int) -> int:
    return k % N

def H1(k: int) -> int:
    return k % L

def task_2():
    # 2) метод лінійних проб (H.i = (H.i-1 + c) % N)
    """
    H0 = H(k)

    Hi = (H0 + i) % N, i = [1, ..., N-1]
    OR
    Hi = (H(i-1) + c) % N

    c - константа, така що c та N взаємно прості
    i - індекс повторного хешування
    """

    N = N1
    O_list = []
    kolizion = []
    d = {}

    tlength = int(math.sqrt(max(Table1_Array)))+2

    i = 0
    for k in range(tlength):
        d[i] = 0
        i += 1

    # Вставка
    for k in Table1_Array:
        hX = H(k)
        count = 1
        while hX in kolizion:
            hX = H(hX) + 2
            count += 1
            if count > tlength:
                break

        kolizion.append(hX)

```

```

d[hX] = k
O_list.append(count)

```

```

print("H = ", kolizion)
print("O = ", O_list)

```

```

pprint(d)
print("Si = ", sum(O_list) / len(O_list))
print("Alpha = ", round(len(O_list)/N, 4))

```

```

# Пошук
Oi = O_list
print("Sf = ", sum(Oi) / len(Oi))

```

```

def task_3():
    # 3) метод квадратичних проб
    """
    H0 = H(k)

    Hi = (H0 + i**2) % N, i > 0
    OR
    Hi = (H0 + c1*i + c2*i) % N

    i - індекс повторного хешування
    """
    N = N2
    O_list = []
    kolizion = []
    d = {}

    i = 0
    for k in range(int(math.sqrt(max(Table1_Array)))+3):
        d[i] = 0
        i += 1

    # Вставка
    I = 1
    for k in Table1_Array:
        hX = H(k)
        count = 1
        while hX in kolizion:
            hX = (H(hX) + 1 ** 2) % N

```

```

        count += 1

        kolizion.append(hX)
        d[hX] = k
        O_list.append(count)

    print("H = ", sorted(kolizion))
    print("O = ", O_list)

    # print(d)
    pprint(d)

    print("Si = ", sum(O_list) / len(O_list))
    print("Alpha = ", round(len(O_list)/N, 4))

    # Пошук
    Oi = O_list
    print("Sf = ", sum(Oi) / len(Oi))

def task_4():
    """
    Використовується 2 функції:
         $H(k) = q(k) \% N$  - вихідна хеш-функція
         $H1(k) = q(k) \% L$  - додаткова хеш функція
        N та L вибираються взаємно простими

    Метод подвійного хешування:
         $H0 = H(k) = q(k) \% N$ 
         $H_i = (H(k) + i * H1(k)) \% N, i > 0$ 
        i - індекс повторного хешування
    """
    # 4) метод подвійного хешування
    N = N3

    # example data
    O_list = []
    kolizion = []
    d = {}

    i = 0
    for k in range(int(math.sqrt(max(Table1_Array)))+3):
        d[i] = 0
        i += 1

```



```

# Вставка
I = 1
kol_repeat = False
for k in Table1_Array:
    hX = H(k)
    count = 1
    while hX in kolizion:
        if H(hX) not in kolizion:
            hX = H(hX)
        else:
            hX = (H(k) + I*H1(k)) % N
            count += 1
            I += 1
    if hX in kolizion:
        kol_repeat = True
        break

    if not kol_repeat:
        kolizion.append(hX)
        d[hX] = k
        O_list.append(count)
        kol_repeat = False

print("H = ", sorted(kolizion))
print("O = ", O_list)

# print(d)
pprint(d)

print("Si = ", sum(O_list) / len(O_list))
print("Alpha = ", round(len(O_list)/N, 4))

# Помык
Oi = O_list
print("Sf = ", sum(Oi) / len(Oi))

def print_decorator(text, func):
    print(text)
    func()
    print()

```

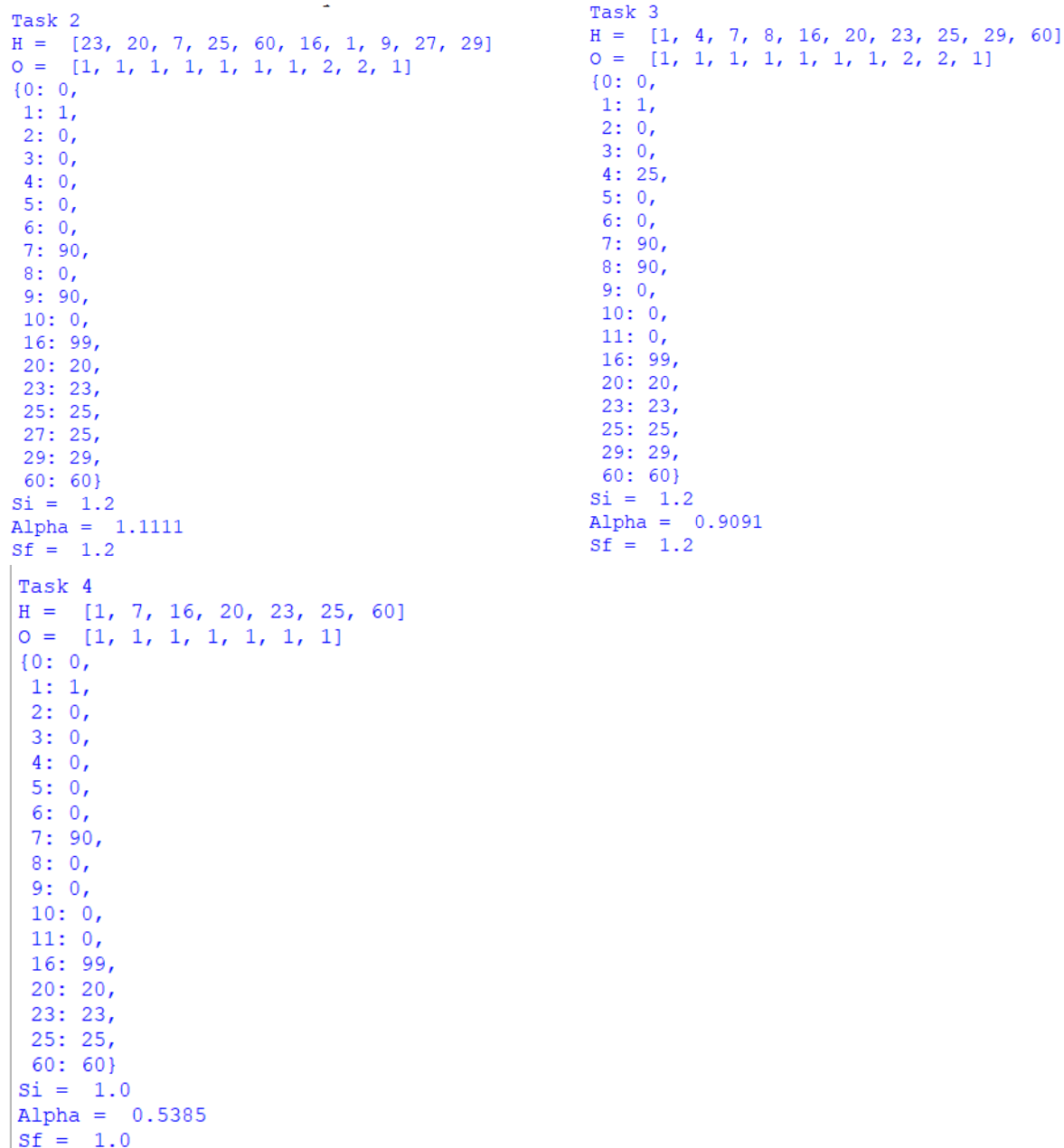
```

print_decorator("Task 2", task_2)
print_decorator("Task 3", task_3)
print_decorator("Task 4", task_4)

```

## Частина 5. Тестування

### Скриншот тестування:



```

Task 2
H = [23, 20, 7, 25, 60, 16, 1, 9, 27, 29]
O = [1, 1, 1, 1, 1, 1, 1, 2, 2, 1]
{0: 0,
 1: 1,
 2: 0,
 3: 0,
 4: 0,
 5: 0,
 6: 0,
 7: 90,
 8: 0,
 9: 90,
10: 0,
16: 99,
20: 20,
23: 23,
25: 25,
27: 25,
29: 29,
60: 60}
Si = 1.2
Alpha = 1.1111
Sf = 1.2

Task 3
H = [1, 4, 7, 8, 16, 20, 23, 25, 29, 60]
O = [1, 1, 1, 1, 1, 1, 1, 2, 2, 1]
{0: 0,
 1: 1,
 2: 0,
 3: 0,
 4: 25,
 5: 0,
 6: 0,
 7: 90,
 8: 90,
 9: 0,
10: 0,
11: 0,
16: 99,
20: 20,
23: 23,
25: 25,
29: 29,
60: 60}
Si = 1.2
Alpha = 0.9091
Sf = 1.2

Task 4
H = [1, 7, 16, 20, 23, 25, 60]
O = [1, 1, 1, 1, 1, 1, 1]
{0: 0,
 1: 1,
 2: 0,
 3: 0,
 4: 0,
 5: 0,
 6: 0,
 7: 90,
 8: 0,
 9: 0,
10: 0,
11: 0,
16: 99,
20: 20,
23: 23,
25: 25,
60: 60}
Si = 1.0
Alpha = 0.5385
Sf = 1.0

```

Рисунок 2 – скриншот тестування

## Висновки

Під час цієї лабораторної роботи я вивчив матеріал лабораторної роботи у розмірі конспекту і додаткових матеріалів з інтернеті та виконав поставлене завдання.