Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра Автоматизованих Систем Обробки Інформації та Управління

Лабораторна робота № 5

з дисципліни «Теорія алгоритмів»

на тему:

**" Метод сортування "**

Виконала:

студентка гр. ІС-02

Павлущенко Ольга

Викладач:

Новікова П. А.

Київ – 2020

**Завдання**

Відсортувати числа, які записані у десятковій системі та які мають d розрядів. Процес сортування виконувати за розрядами цих чисел. в Реалізувати алгоритм RadixSort.

Нехай є n d-значних чисел, в яких кожна цифра приймає одне з k можливих значень. Розробити алгоритм RadixSort, який дозволить виконати коректне сортування цих чисел за час Θ(d(n + k)). Використовувати стійке сортування, яке використовуються цим алгоритмом, який має час роботи Θ(n + k).Надати можливість вибору числа d.

**Програмний код**

import random

import math

def main():

print ("Enter the digit number: ")

d = int(input()) #кількість розрядів

p = pow(10,d-1)

r = pow(10,(d))

n = 7

A = [random.randint(p,r) for i in range(n)] #масив чисел

print (A)

RadixSort(A,d)

print ("Sorted: ", A)

def CountingSort(A,k):

C = [0 for i in range(10)] #додатковий масив для зберігання кількості елементів

B = [0 for i in range(len(A))] #масив відсортованих елементів

for j in range(len(A)):

C[int((A[j]/k)%10)] += 1 #кількість кожного з елементів з даним значенням розряду

for i in range (1,10): #визначаємо індекси елементів масива, на які вони мають стати у відсоротованому масиві

C[i] += C[i-1]

for j in range (len(A)-1,-1,-1):

B[C[int((A[j]/k)%10)]-1] = A[j] #поміщення значень в вихідний масив

C[int((A[j]/k)%10)] -= 1 #зменшуємо кількість вхідних елементів на 1;

#оскільки може бути ще такий елемент і тоді він стане попереду

for p in range(len(B)):

A[p] = B[p]

def RadixSort(A,d):

digitNumber = 1 #розряд числа (від молодшого до старшого)

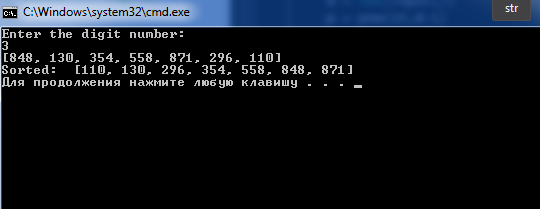
for i in range(d): #проходи по розрядам

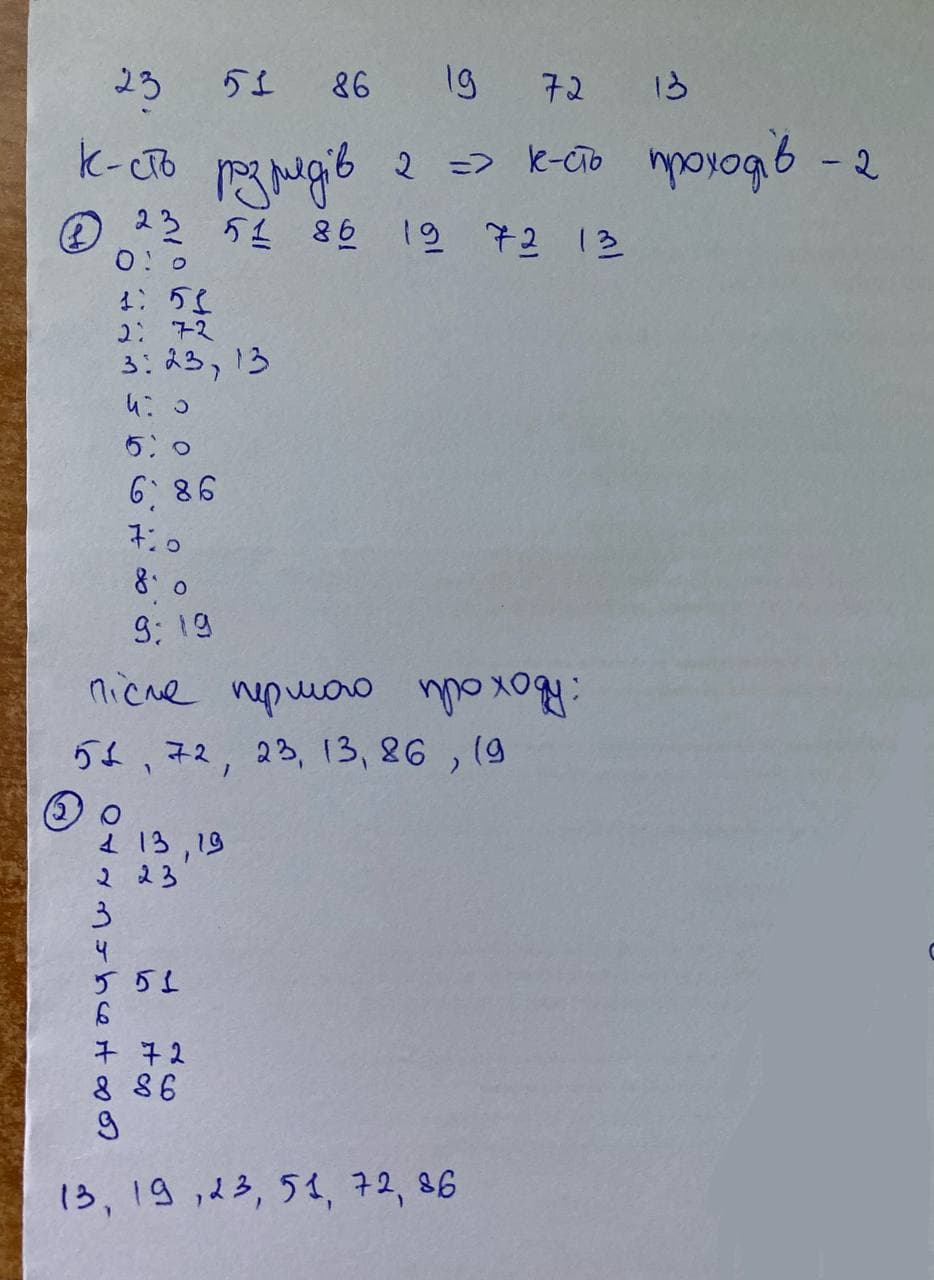
CountingSort(A,digitNumber)

digitNumber \*= 10 #перехід на наступний розряд

return A

main()

**Результати**

****

**Висновок**

У ході лабораторної роботи було реалізовано RadixSort, в основі якого лежить модернізоване сортування підрахунком(CountingSort). Складність даного алгоритму становить Θ(d(n+k)).

Працює таким чином, що числа вхідного масиву сортуються по розряду, починаючи з молодшого, за допомогою CountingSort.

Основна ідея CountingSort полягає в тому, щоб визначити кількість елементів, які менше х, за допомогою цієї інформації х можна розмістити на тій позиції вихідного масиву, де він має знаходитися, коли ж декілька елементів мають одне і те саме значення, то: поміщуючи елемент у потрібну позицію вихідного масиву, ми зменшуємо кількість елементів, які не перевищують поточний на 1, таким чином, якщо є наступний елемент із таким самим значенням, то він розміщується перед елементом з таким ж значенням. Його є сенс використовувати, коли k співрозмірне з n або менше, але ні в якому разі не перевищує n в кілька разів. Наприклад, при k = O(n), складність CountingSort дорівнює Θ(n).

Цей алгоритм не засновується на порівняннях та є стійким, тобто він не змінює порядок елементів з однаковим значенням.