

Модульна контрольна робота №2

Маньківський Владислав

BT-21-1[2]

ВАРІАНТ 2

СОРТУВАННЯ ЗА РОЗРЯДАМИ (RADIX SORT)

Мета : реалізація та опис алгоритма сортування за розрядами.

2.1 Хід роботи

Звіт повинен містити: назву алгоритму; короткий опис алгоритму із зазначенням складності; структури даних, які будуть використані; тестові дані для програми; результати роботи програми (поясненням зі скрінами); посилання на програму в гітлаб.

2. Сортування за розрядами (radix sort)

2.1.1

Завдання:

Результати роботи програми:

На скріншоті можна побачити, що алгоритм правильно відсортував масив з 10 елементів, та тестування різної кількості елементів. Можна чітко побачити, що час сортування на скріншотах (Рис 2.1) відрізняється певною похибкою, чим більше елементів, тим більше похибка.

					ДУ«Житомирська політехніка».21.121.02.000–Мкр 2						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Маньківський В.В			Звіт з модульної контрольної роботи			Літ.	Арк.	Аркушів	
Перевір.		Петросян Р.В.							1	9	
Керівник								ФІКТ Гр. ВТ-21-1[2]			
Н. контр.											
Зав. каф.											

Сортування за розрядами	Сортування за розрядами
Вхідні дані: 100 91 98 24 3 1 65 13 90 56	Вхідні дані: 47 22 56 52 71 79 22 23 61 86
Вихідні дані: 1 3 13 24 56 65 90 91 98 100	Вихідні дані: 22 22 23 47 52 56 61 71 79 86
10 -> 0,0132 мс	10 -> 0,0198 мс
50 -> 0,03 мс	50 -> 0,0404 мс
100 -> 0,0574 мс	100 -> 0,1008 мс
500 -> 0,2662 мс	500 -> 0,4224 мс
1000 -> 0,4977 мс	1000 -> 0,6977 мс
5000 -> 3,5529 мс	5000 -> 4,1691 мс
10000 -> 5,7124 мс	10000 -> 22,0059 мс
50000 -> 25,3734 мс	50000 -> 34,8301 мс
100000 -> 57,0125 мс	100000 -> 70,4563 мс
500000 -> 275,2921 мс	500000 -> 485,27 мс
1000000 -> 746,8525 мс	1000000 -> 615,2987 мс
5000000 -> 2609,6576 мс	5000000 -> 2964,3452 мс
10000000 -> 4836,1401 мс	10000000 -> 6755,1663 мс
50000000 -> 24324,6685 мс	50000000 -> 25509,4784 мс
100000000 -> 48953,7353 мс	100000000 -> 48768,4373 мс
500000000 -> 277066,3902 мс	500000000 -> 254160,0113 мс
1000000000 -> 661559,2402 мс	1000000000 -> 637938,3784 мс

Рисунок 2.1 – Результати виконання завдання

Загальний словесний опис алгоритму:

Сортування за розрядами (radix sort) – швидкий стабільний алгоритм сортування даних, який використовує розряди числа. Головна ідея алгоритму, сортувати за кожним розрядом, від наймолодшого до найстаршого. Плюс алгоритма – швидкість завжди стала. Мінус в тому, що він сортує лише цілі числа. Складність алгоритму залежить лише від кількості елементів, чим їх більше, тим довше сортує.

Загальна складність роботи алгоритму з використанням сортування підрахунком є $O(D * (N + K))$ (де N — кількість елементів в масиві; K — кількість символів у алфавіті, якщо впорядковуються десяткові числа, то $K = 10$ D — кількість розрядів). Якщо впорядковувати цим алгоритмом цілі числа, то складність буде $O(N \log M)$, де M — найбільший елемент масиву.

		Маньківський В.М.			ДУ«Житомирська політехніка».21.121.02.000–Мкр 2	Арк.
		Петросян Р.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Короткий опис алгоритму:

1. Пошук найбільшого елемента.
2. Розрахунок кількості ітерацій розряду в найбільшому елементі, входження до циклу починаючи з найменшого.
3. Створення масиву для вихідних даних, та додаткового масиву для оперування над елементами.
4. Заповнення додаткового масиву '0'.
5. Пошук кількості цифр певного розряду від 0 до 9.
6. Перерозподіл цифр в додатковому масиві під їх фактичне розташування.
7. Встановлення елементів на їх відсортоване місце одразу в масиві вихідних даних.
8. Копіювання до основго масиву.

		Маньківський В.М			ДУ«Житомирська політехніка».21.121.02.000–Мкр 2	Арк.
		Петросян Р.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.2 – Блок-схема до алгоритму

Структура даних:

Було вирішено використати структуру даних типу `int`, заповнену числами від 0 до 100 для демонстрації, від 0 до 2147483646 для тестування, так як алгоритм не підтримує числа з плаваючою комою. Структур кілька, так як необхідно було переглянути результати сортування за часом, а саме: `arr` для демонстрації сортування, всі інші, для тестування, `arr_10`, `arr_50`, `arr_100`, `arr_500`,

arr_1000, arr_5000, arr_10000, arr_50000, arr_100000, arr_500000, arr_1000000, arr_5000000, 10000000, ще окремо були використані структури для дуже великої кількості чисел, arr_50000000, arr_100000000, arr_500000000, arr_1000000000.

Для самого алгоритму було використано arr – основний масив, output – для кінцевого розміщення елементів під час ітерації, count – для оперування над розміщенням чисел.

Тестові дані для програми:

Було перевірено алгоритм на такому діапазоні чисел [0; 2147483646].

Було перевірено алгоритм на такій кількості елементів: 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000, 500000, 1000000, 5000000, 10000000, 50000000, 100000000, 500000000, 1000000000.

Графіки та таблиця:

		Маньківський В.М			ДУ«Житомирська політехніка».21.121.02.000–Мкр 2	Арк.
		Петросян Р.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Кількість елементів	Розрядами
	масив (мс)
10	0,0132
50	0,03
100	0,0574
500	0,2662
1000	0,4977
5000	3,5529
10000	5,7124
50000	25,3734
100000	57,0125
500000	275,2921
1000000	746,8525
5000000	2609,6576
10000000	4836,1401
50000000	24324,6685
100000000	48953,7353
500000000	277066,3902
1000000000	661559,2402

Рисунок 2.3 – Таблиця



Рисунок 2.4 – Графік від 10 до 10000

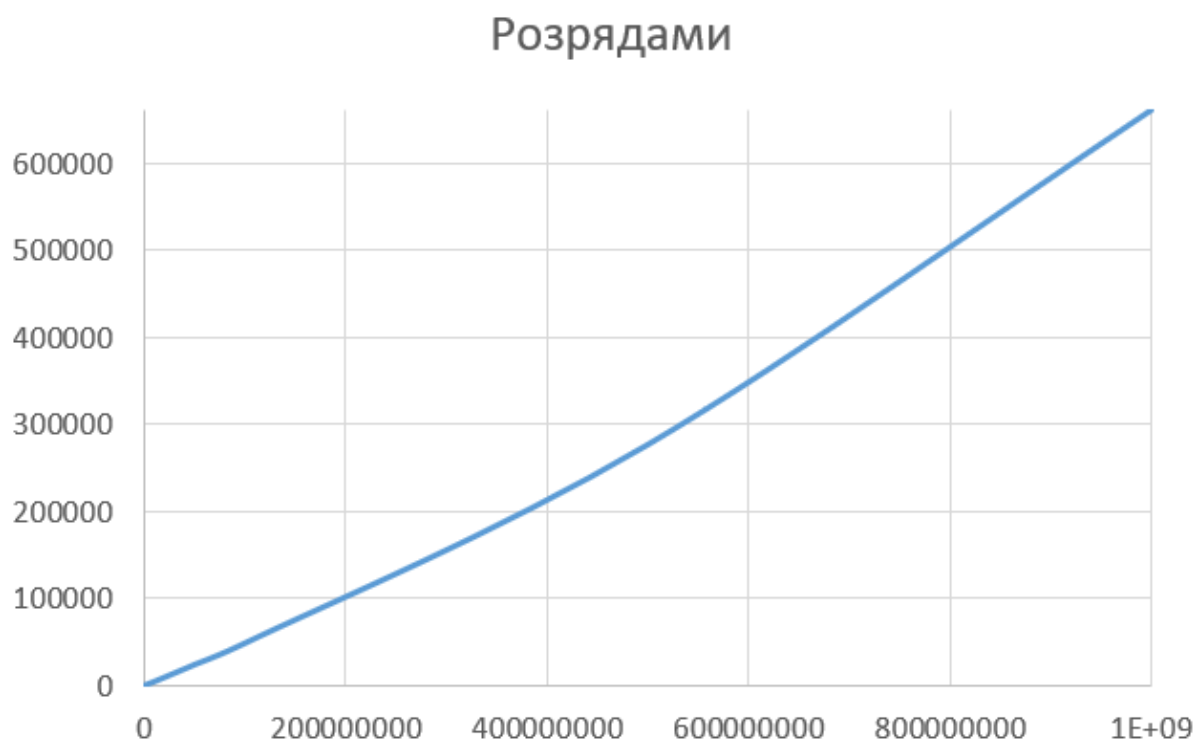


Рисунок 2.5 – Графік від 10 до 10000000000

Посилання на програму в гітлаб:

https://gitlab.com/RAK_MANIAC/algorithmi_modular_control_work_2

Лістинг:

```
using System.Diagnostics;
namespace Radix_Sort
{
    class Program
    {
        public static int Max(int[] arr, int size)//знаходження максимального елемента
        {
            int max = arr[0];
            for (int i = 1; i < size; i++)
                if (arr[i] > max)
                    max = arr[i];
            return max;
        }
        public static void RadixSort(int[] arr, int size)//сортування за розрядами
        {
            int max = Max(arr, size);//максимальний елемент
            for (int exponent = 1; max / exponent > 0; exponent *= 10)//кількість
ітерацій відповідна кількості розрядів максимального елемента
                CountingSort(arr, size, exponent);//сортування
        }
        public static void CountingSort(int[] arr, int size, int exponent)
        {
            int[] output = new int[size];//вихідний масив
            int[] count = new int[10];//масив для оперування над елементами
            for (int i = 0; i < 10; i++)//заповнення додаткового масива значенням '0'
                count[i] = 0;
            for (int i = 0; i < size; i++)//кількість цифр певного розряду від 0 до 9
                count[(arr[i] / exponent) % 10]++;
            for (int i = 1; i < 10; i++)//фактичне розміщення цифр
                count[i] += count[i - 1];
            for (int i = size - 1; i >= 0; i--)//встановлення елементів на їх
відсортоване за розрядом місце
            {
                output[count[(arr[i] / exponent) % 10] - 1] = arr[i];
                count[(arr[i] / exponent) % 10]--;
            }
            for (int i = 0; i < size; i++)//копіювання в основний масив
                arr[i] = output[i];
        }
        public static void check(int[]arr, int k, int min, int max)
        {
            Stopwatch stopwatch = new Stopwatch();
            Random random = new Random();
            for (int i = 0; i < k; i++)
                arr[i] = random.Next(min, max + 1);
            stopwatch.Start();
            RadixSort(arr, k);
            stopwatch.Stop();
            Console.WriteLine($"{k} -> {stopwatch.Elapsed.TotalMilliseconds} мс");
            stopwatch.Reset();
        }

        static void Main()
        {
            Console.OutputEncoding = System.Text.Encoding.Unicode;
            Console.InputEncoding = System.Text.Encoding.Unicode;
        }
    }
}
```

		Маньківський В.М			ДУ«Житомирська політехніка».21.121.02.000–Мкр 2	Арк.
		Петросян Р.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

Random random = new Random();

Console.WriteLine("Сортування за розрядами");
int size = 10, min = 0, max = 100;
//k >= 1, min >= 0, max <= 2147483646;
int[] arr = new int[size];
for (int i = 0; i < size; i++)
    arr[i] = random.Next(min, max + 1);
Console.WriteLine("\nВхідні дані: ");
foreach (var x in arr)
{
    Console.Write($" {x}");
}
RadixSort(arr, size);
Console.WriteLine("\nВихідні дані: ");
foreach (var x in arr)
{
    Console.Write($" {x}");
}
Console.WriteLine("\n");

////////////////////////////////////

min = 0;
max = 2147483646;

size = 10;
int[] arr_10 = new int[size];
check(arr_10, size, min, max);

size = 50;
int[] arr_50 = new int[size];
check(arr_50, size, min, max);

size = 100;
int[] arr_100 = new int[size];
check(arr_100, size, min, max);

size = 500;
int[] arr_500 = new int[size];
check(arr_500, size, min, max);

size = 1000;
int[] arr_1000 = new int[size];
check(arr_1000, size, min, max);

size = 5000;
int[] arr_5000 = new int[size];
check(arr_5000, size, min, max);

size = 10000;
int[] arr_10000 = new int[size];
check(arr_10000, size, min, max);

size = 50000;
int[] arr_50000 = new int[size];
check(arr_50000, size, min, max);

size = 100000;
int[] arr_100000 = new int[size];
check(arr_100000, size, min, max);

size = 500000;

```

		Маньківський В.М			ДУ«Житомирська політехніка».21.121.02.000–Мкр 2	Арк.
		Петросян Р.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

int[] arr_500000 = new int[size];
check(arr_500000, size, min, max);

size = 1000000;
int[] arr_1000000 = new int[size];
check(arr_1000000, size, min, max);

size = 5000000;
int[] arr_5000000 = new int[size];
check(arr_5000000, size, min, max);

size = 10000000;
int[] arr_10000000 = new int[size];
check(arr_10000000, size, min, max);

////
//      size = 50000000;
//      int[] arr_50000000 = new int[size];
//      check(arr_50000000, size, min, max);

//      size = 100000000;
//      int[] arr_100000000 = new int[size];
//      check(arr_100000000, size, min, max);

//      size = 500000000;
//      int[] arr_500000000 = new int[size];
//      check(arr_500000000, size, min, max);

//      size = 1000000000;
//      int[] arr_1000000000 = new int[size];
//      check(arr_1000000000, size, min, max);
    }
}

```

Висновки: під час цієї модульної контрольної роботи я вивчив новий метод сортування, створив його алгоритм та дослідив. Під час виконання описав алгоритм та результати програми, словами та блок-схемою, зобразив результати програми.

		Маньківський В.М			ДУ«Житомирська політехніка».21.121.02.000–Мкр 2	Арк.
		Петросян Р.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		