

Лабораторна робота

«Сортування»

з дисципліни «Структури даних, аналіз і алгоритми коп'ютерної
обробки інформації»

Виконала: студентка групи ІПЗ-11

Манойлова Катерина Борисівна

1. Умова задачі

Написати програму мовою C# , реалізувавши алгоритм сортування злиттям. Продемонструвати роботу (ефективність, час виконання) програм на різних структурах даних (масив, лінійний зв'язаний список). Навести аналіз отриманих результатів.

2. Аналіз задачі

Алгоритми сортування розроблятимемо для двох структур даних: масив та лінійний зв'язаний список. Для зручності структури даних будуть зберігати цілі числа.

Програма буде генерувати масив та заповнювати згенерованими у ньому значеннями лінійний зв'язний список, розмір та діапазон значень вводяться з консолі. Після сортування виводиться таблиця порівняння часу дії алгоритмів для списку та масиву.

3. Структура основних вхідних та вихідних даних

Вхідними даними алгоритмів є відповідна структура даних (масив або лінійний зв'язний список) цілих чисел.

Вихідними даними є та сама структура даних, відсортована за зростанням.

4. Алгоритм розв'язання задачі

1) Лінійний пошук:

Для масиву:

```
left = lowI;
right = midI + 1;
tempArr = new int[highI - lowI + 1];
index = 0;

while ((left <= midI) && (right <= highI))
{
    if (Arr[left] < Arr[right])
    {
        tempArr[index] = Arr[left];
        left++;
    }
    else
    {
        tempArr[index] = Arr[right];
```

```

        right++;
    }

    index++;
}

for (var i = left; i <= midI; i++)
{
    tempArr[index] = Arr[i];
    index++;
}

for (var i = right; i <= highI; i++)
{
    tempArr[index] = Arr[i];
    index++;
}

for (var i = 0; i < tempArr.Length; i++)
{
    Arr[lowI + i] = tempArr[i];
}

```

2) Пошук з бар'єром

Для масиву:

```

Merge(Arr, lowI, midI, highI)
{
    left = lowI;
    right = midI + 1;
    tempArr = int[highI - lowI + 1];
    index = 0;

    while ((left <= midI) && (right <= highI))
    {
        if (Arr[left] < Arr[right])
        {
            tempArr[index] = Arr[left];
            left++;
        }
        else
        {
            tempArr[index] = Arr[right];
            right++;
        }

        index++;
    }
}

```

```

    }

    for (var i = left; i <= midI; i++)
    {
        tempArr[index] = Arr[i];
        index++;
    }

    for (var i = right; i <= highI; i++)
    {
        tempArr[index] = Arr[i];
        index++;
    }

    for (var i = 0; i < tempArr.Length; i++)
    {
        Arr[lowI + i] = tempArr[i];
    }
}

MergeSort(Arr, lowI, highI)
{
    if (lowI < highI)
    {
        midI = (lowI + highI) / 2;
        MergeSort(Arr, lowI, midI);
        MergeSort(Arr, midI + 1, highI);
        Merge(Arr, lowI, midI, highI);
    }

    return Arr;
}

```

Для списка:

```

Node Merge(Node a, Node b)
{
    Node result = null;

    if (a == null)
    {
        return b;
    }
    if (b == null)
    {
        return a;
    }
}

```

```

    if (a.Data <= b.Data)
    {
        result = a;
        result.Next = Merge(a.Next, b);
    }
    else
    {
        result = b;
        result.Next = Merge(a, b.Next);
    }

    return result;
}

Node mergeSort(Node h)
{
    if (h == null || h.Next == null)
    {
        return h;
    }

    Node Mid = LinkedList.findMid(h);
    Node NextMid = Mid.Next;

    Mid.Next = null;

    Node left = mergeSort(h);

    Node right = mergeSort(NextMid);

    Node sortedlist = Merge(left, right);
    return sortedlist;
}

```

5. Текст програми

Текст програми у репозиторії GitHub за посиланням:

https://github.com/ManoilovaKaterina/Lab_Report/tree/main/%D0%90%D0%A1%D0%94/SortLab

6. Набір тестів

Тест 1:

10 елементів:

0 38 3 26 8 52 77 11 18 31

Тест 2:

1000 елементів:

72 87 52 35 48 1 83 32 83 21 80 13 59 85 41 3 2 12 94 32 48 45 73 24 87 20 3 20 6
75 24 87 42 59 25 2 77 41 38 86 64 69 49 42 14 58 7 69 69 45 37 39 53 54 25 26 13
28 47 27 98 63 26 7 96 92 70 0 60 39 26 60 73 8 68 78 96 31 9 29 13 33 50 61 38 85
33 88 33 98 88 48 93 10 87 1 43 42 45 22 87 6 9 30 19 76 65 45 86 47 30 82 19 18
14 65 13 64 69 11 59 82 66 61 51 77 67 62 21 66 34 53 85 86 42 6 24 20 41 61 20 88
1 86 67 5 29 75 96 22 88 79 73 34 63 4 40 47 78 42 9 2 23 19 12 76 97 32 76 8 40 92
22 7 91 70 81 79 94 45 47 91 66 99 78 55 80 50 23 37 65 77 41 99 51 17 64 82 74 90
8 97 98 88 81 95 56 65 43 92 23 60 52 33 94 17 35 24 40 57 96 46 8 38 42 62 51 23
55 73 6 99 5 4 37 50 92 78 17 83 98 15 7 31 14 4 25 7 5 33 81 39 41 17 94 61 88 60
45 19 43 33 9 69 85 23 55 48 96 43 24 57 7 57 58 80 38 77 23 37 37 43 17 22 92 67
57 87 9 75 62 32 33 97 39 65 5 38 45 90 49 76 11 62 8 24 32 84 59 13 23 11 36 8 82
0 15 87 77 18 66 68 38 20 81 92 24 9 17 92 75 0 32 32 88 60 32 55 13 67 34 73 28
96 52 50 95 24 15 39 49 18 60 26 24 80 38 90 81 16 0 22 67 66 38 22 78 3 19 21 67
60 74 9 83 92 39 41 68 30 96 99 93 77 43 56 39 5 8 7 21 41 74 96 67 12 6 62 58 29
71 91 5 94 71 89 43 50 42 31 41 96 21 50 19 0 29 89 23 81 83 73 95 11 99 26 86 77
42 71 86 76 83 39 58 4 93 99 5 53 13 88 63 76 65 24 20 23 76 66 83 16 39 76 45 98
96 93 94 45 2 65 8 71 45 64 13 82 92 14 7 29 83 69 68 94 9 18 46 74 6 62 0 76 88 69
37 6 93 59 8 0 5 60 51 47 80 92 31 1 11 37 31 62 59 53 33 69 8 51 88 77 47 19 39 40
64 31 82 75 26 6 88 33 6 6 24 23 18 20 13 17 87 44 63 68 31 38 17 35 36 68 98 42
70 30 52 85 20 11 82 48 9 56 74 5 49 98 55 52 28 10 51 87 37 71 90 2 56 71 8 26 14
46 39 58 8 46 62 75 53 39 2 7 38 65 7 31 70 57 18 33 83 64 7 25 16 10 4 99 40 50 29
49 2 56 33 62 17 16 97 3 35 79 98 88 7 44 49 72 63 59 32 98 52 74 43 50 39 13 41
97 41 62 35 3 9 53 4 82 31 45 13 54 60 15 98 3 66 18 18 72 61 32 36 81 17 31 96 27
12 83 23 4 74 99 61 73 44 94 79 54 39 66 41 17 45 78 38 37 76 39 83 21 94 69 36 9
26 54 85 78 57 76 69 47 22 9 80 61 53 24 58 71 39 64 32 94 91 18 35 39 93 59 51 73
0 2 9 4 62 52 46 90 9 1 97 63 97 93 95 35 98 76 83 51 81 12 81 30 36 42 18 7 18 45
85 97 25 96 47 19 99 75 98 1 77 68 62 38 67 69 97 97 22 99 41 16 75 99 92 87 20 79
68 20 34 39 71 63 16 99 37 0 46 75 35 22 78 82 73 12 50 43 62 67 44 20 84 18 3 68
22 25 4 59 99 20 7 78 67 38 97 98 22 68 31 96 50 47 63 19 37 93 25 79 37 76 16 0
75 13 55 52 59 47 76 12 96 87 76 15 15 0 15 35 14 51 21 94 35 47 70 37 55 84 30 29
99 24 21 23 3 6 2 53 82 45 38 74 91 19 0 63 70 48 3 37 77 90 64 85 54 21 65 27 43
18 97 75 16 47 12 71 65 52 11 9 98 61 52 69 13 47 3 16 47 6 66 6 81 93 51 9 59 36
68 82 41 74 10 64 47 63 58 43 6 29 98 17 36 93 39 16 37 95 72 7 18 48 80 36 51 90
93 22 96 3 11 29 84 29 68 23 50 87 21 50 88 59 10 18 7 48 70 88 53 35 71 64 63 61
63 92 93 73 13 57 69 35 47 3 17 69 33 7 24 15 65 50 73 22 18 58 38 62 29 3 20 5

7. Результати тестування програми та аналіз отриманих помилок

Тест 1:

```

----- Порівняння сортування злиттям масива та списку: -----
Введіть розмір структур даних: 10
Введіть діапазон значень: 100

Згенерований масив:
0 38 3 26 8 52 77 11 18 31
Згенерований список:
0 38 3 26 8 52 77 11 18 31

Відсортований масив:
0 3 8 11 18 26 31 38 52 77
Відсортований список:
0 3 8 11 18 26 31 38 52 77

=====Час виконання=====
Для масиву:      00:00:00.0000052
Для списку:      00:00:00.0000082
=====

```

Тест 2:

[illegible]

Висновок:

Час виконання алгоритму напряму залежить від розміру структури даних. Сортування виконується на масиві швидше, ніж на лінійному зв'язному списку.