Лабораторна робота №2

Виконала: студентка групи ІПЗ-11

Сирота Ангеліна Олександрівна

* Умова задачі:

Написати програму мовою C#, що реалізує алгоритм сортування злиттям. Продемонструвати роботу (ефективність, час виконання) програм на різних структурах даних (масив, лінійний зв’язаний список), з різними умовами, що забезпечують зменшення часу виконання. Навести аналіз отриманих результатів.

* Аналіз задачі:

Генеруємо масив цілих чисел, з яких потім створюємо лінійний список.

Алгоритм пошуку у масиві реалізовуємо в окремому класі. Для зручності, алгоритми пошуку у списку реалізовуємо у окремому класі зі списками.

Щоб виміряти час роботи алгоритму використовуємо клас Stopwatch(), використовуючи простір імен System.Diagnostics.

* Структура основних вхідних та вихідних даних:

Вхідними даними є масив цілих чисел. З елементів масиву ми створюємо лінійний зв'язний список.

Вихідними даними є масив і список, що містять в собі вже відсортовану послідовність елементів.

* Алгоритм розв'язання задачі:
* ***Сортування злиттям у масиві:***

1. Функція, що ділить основний масив на підмасиви:

int LeftLength = mas.Length / 2;

int RightLength = mas.Length – LeftLength;

int[] mas;

int[] LeftArr;

int[] RightArr;

int j = LeftLength;

for (int i = 0; i < LeftLength; i++)

LeftArr[i] = mas[i];

for (int i = 0; i < RightLength; i++)

{

RightArr[i] = mas[j];

j++;

}

MergeSort (LeftArr); // рекурсивний виклик функції

MergeSort (RightArr);

Merge (mas, LeftArr, RightArr); // функція, що порівнює значення з лівого і правого масиву і записує впорядковані значення у mas.

1. Функція, що порівнює значення з лівого і правого масиву і записує впорядковані значення у mas:

int[]mas;

int[] left;

int[] right;

int LIndex = 0, RIndex = 0, Index = 0;

while (Index < mas.Length)

{

if (LIndex >= left.Length)

{

mas[Index] = right[RIndex];

RIndex++;

}

else if (RIndex >= right.Length)

{

mas[Index] = left[LIndex];

LIndex++;

}

else if (left[LIndex] < right[RIndex])

{

mas[Index] = left[LIndex];

LIndex++;

}

else

{

max[Index] = right[RIndex];

RIndex++;

}

Index++;

}

* *Сортування злиттям у списку:*
  + Функція, що рекурсивно розбиває список на 2 частини:

Item middle = getMiddle (head, end); // знайти середній елемент списку

Item NextMiddle = middle.Next;

middle.Next = null;

Item left = MergeSort (head);

Item right = MergeSort (NextMiddle);

Item ResultList = Merge (left, right);

return ResultList;

* + Функція, що порівнює значення з першого і другого списку і записує впорядковані значення у результуючий список:

Item result = null;

Item Left;

Item Right;

if (Left == null)

return Right;

if (Right == null)

return Left;

if (Left.Data <= Right.Data)

{

result = Left;

result.Next = Merge (Left.Next, Right)

}

else

{

result = Right;

result.Next = Merge (Left, Right.Next)

}

return result;

* + Функція, що знаходить середній елемент у списку:

Item current = head;

int k = 0, k1 = 0;

while (current != null)

{

k++;

current = current.Next;

}

current = head;

while (k1\*2 + 1 < k - 1)

{

k1++;

current = current.Next;

}

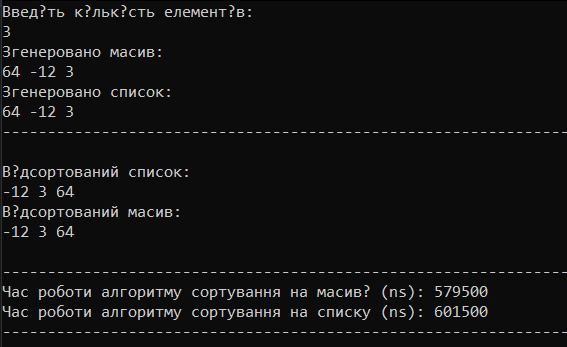
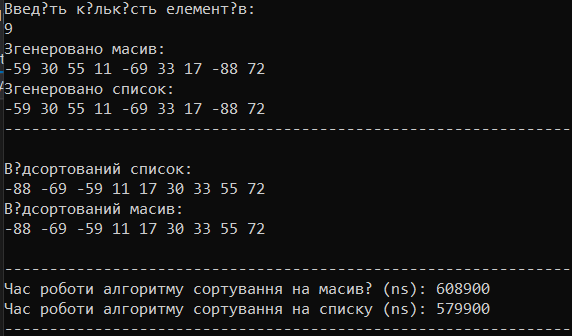
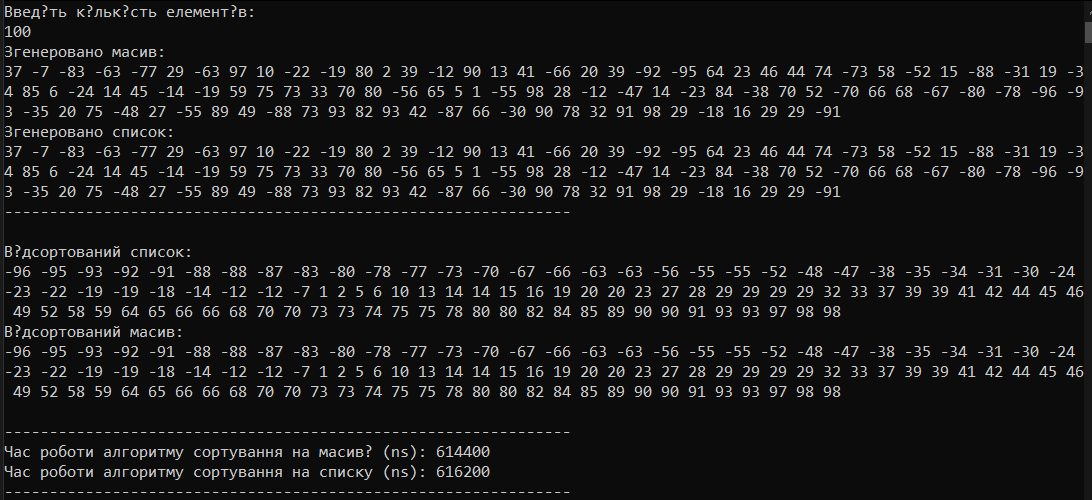
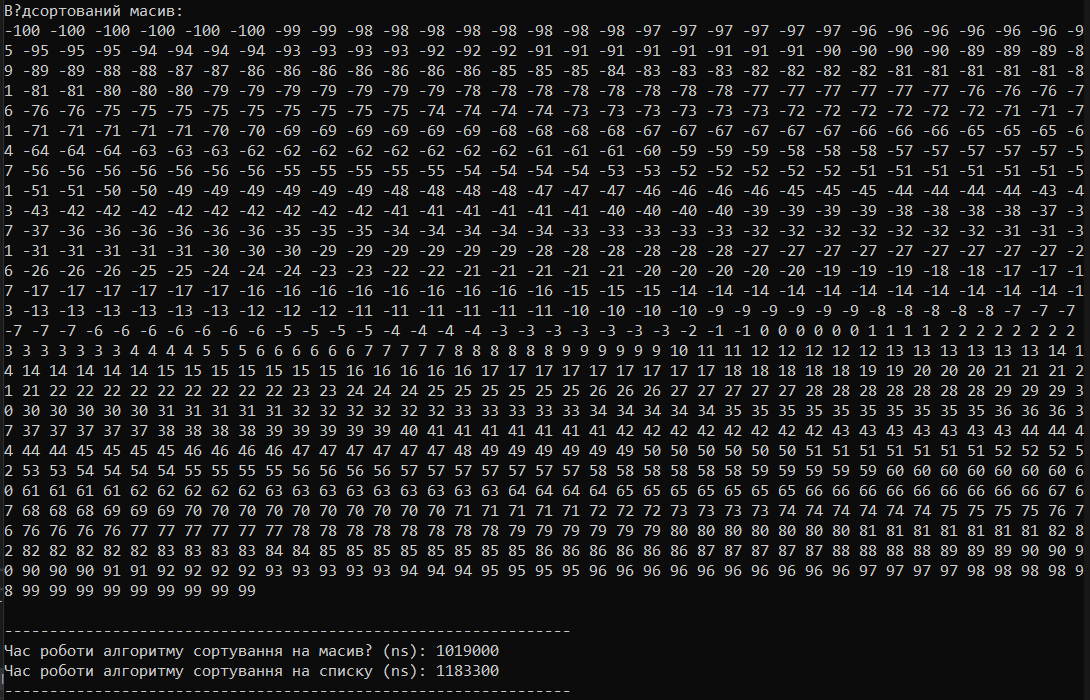
return current;

* Текст програми:

Увесь код програми можна переглянути в репозиторії [github](https://github.com/LinaSyrota/lab_report/tree/main/%D0%90%D0%A1%D0%94/%D0%9B%D0%B0%D0%B1%202).

* Набір тестів:
  + 3 елемента: 64 -12 3
    - * відсортовано: -12 3 64
  + 9 елементів: -59 30 55 11 -69 33 17 -88 72
    - * відсортовано: -88 -69 -59 11 17 30 33 55 72
  + 100 елементів: 37 -7 -83 -63 -77 29 -63 97 10 -22 -19 80 2 39 -12 90 13 41 -66 20 39 -92 -95 64 23 46 44 74 -73 58 -52 15 -88 -31 19 -34 85 6 -24 14 45 -14 -19 59 75 73 33 70 80 -56 65 5 1 -55 98 28 -12 -47 14 -23 84 -38 70 52 -70 66 68 -67 -80 -78 -96 -93 -35 20 75 -48 27 -55 89 49 -88 73 93 82 93 42 -87 66 -30 90 78 32 91 98 29 -18 16 29 29 -91
    - * відсортовано: -96 -95 -93 -92 -91 -88 -88 -87 -83 -80 -78 -77 -73 -70 -67 -66 -63 -63 -56 -55 -55 -52 -48 -47 -38 -35 -34 -31 -30 -24 -23 -22 -19 -19 -18 -14 -12 -12 -7 1 2 5 6 10 13 14 14 15 16 19 20 20 23 27 28 29 29 29 29 32 33 37 39 39 41 42 44 45 46 49 52 58 59 64 65 66 66 68 70 70 73 73 74 75 75 78 80 80 82 84 85 89 90 90 91 93 93 97 98 98
  + 1000 елементів: -56 -16 32 -49 -23 51 -89 2 17 13 -76 17 51 15 31 78 -10 35 -24 -16 25 -88 86 -17 56 -67 49 -15 -24 8 -72 73 -71 -13 -24 31 -81 96 48 -77 32 35 34 -69 47 81 -62 -66 77 -58 -21 1 3 66 -40 -68 -3 -51 -5 -56 -99 25 -75 -92 -16 -49 -12 -36 -61 -100 59 -7 87 -47 51 89 -23 55 35 74 95 14 8 -67 -72 27 -31 -37 -73 -57 -67 69 1 4 -79 -94 9 25 46 -4 -7 74 94 77 -17 85 13 -50 -86 -30 54 41 13 20 -52 -14 -83 92 79 57 81 -75 -64 96 66 89 92 -17 -69 44 -20 52 -91 -61 -19 -79 19 31 49 -57 -87 -81 -52 57 -11 -8 -44 80 -45 35 89 16 46 88 14 -33 37 98 -89 -7 93 -70 -82 -31 90 -98 31 -27 -62 -73 66 -14 -57 36 43 62 76 -21 6 -27 33 -89 -73 -4 55 63 -94 -80 24 -86 -17 21 20 -45 -51 83 -100 79 67 -13 60 82 42 -72 68 3 -27 -91 -55 52 -9 -42 -9 37 60 81 -97 22 13 -73 96 11 49 -76 43 -55 65 87 -14 60 61 97 12 -81 13 50 -31 99 28 -48 -11 96 75 72 22 -75 -71 82 -6 -32 80 -22 -35 39 -9 -51 -67 17 -16 22 37 -14 22 -16 91 -6 45 -29 -76 -91 12 -97 94 -39 37 99 35 76 1 18 -84 83 -8 -82 80 -17 31 72 28 -36 17 52 14 -95 14 -98 -26 -10 81 -72 -21 57 58 -15 56 76 -16 -97 -98 -9 14 -44 -85 66 6 54 93 3 4 7 62 73 43 60 -8 59 -11 88 46 34 25 -56 95 -27 -62 -42 -76 -93 -38 -97 99 27 70 -63 49 -40 36 47 -1 60 -91 70 85 -27 -34 6 -86 98 92 96 -14 -5 -43 -33 66 -31 -40 42 28 -53 30 86 -59 1 8 -66 -97 5 -90 -69 -77 41 -12 85 25 54 -42 22 -93 32 -56 -75 -86 -34 16 3 -30 2 -30 -20 99 93 -81 -48 -70 -78 29 -16 65 -69 2 8 88 61 72 45 -94 -29 41 -89 -90 38 35 95 -98 4 71 74 84 -96 63 0 -65 29 -69 -71 21 -64 55 2 -75 73 57 75 -71 35 -63 -47 -7 71 -3 99 61 -42 -31 -39 -42 -77 -51 26 -49 54 0 60 -33 -98 -27 -42 30 42 -77 70 15 -61 85 -64 81 59 66 -34 -33 18 -20 -88 7 64 39 76 -41 70 70 -3 87 -71 -49 -29 -27 96 7 -74 -38 78 23 51 47 27 -17 22 -27 -68 -21 -2 -43 82 -42 32 -27 -13 -48 3 -37 -4 98 59 71 26 82 -14 88 14 63 85 -48 39 -96 -11 95 29 -35 80 -19 -83 3 -73 -10 -28 -60 -96 70 18 98 85 43 -96 63 65 79 63 79 -79 44 96 34 58 -32 -6 -54 17 78 69 37 43 -78 -28 -79 -54 93 71 -32 -1 -14 77 16 -31 16 78 -59 -53 76 -62 9 -68 50 -56 -64 -16 -17 44 58 96 -75 22 -74 -100 80 -46 62 -29 82 17 71 -93 -6 -21 34 66 -20 -36 -58 -10 -47 99 41 39 -41 5 92 78 -65 36 63 -51 -87 -28 -26 64 50 15 -11 -39 -85 69 -36 -41 -34 2 38 -29 9 -31 -96 -82 2 -32 43 -74 80 -13 40 -25 -38 84 -65 -55 53 22 94 -79 -96 70 -85 65 74 81 -26 -17 34 -18 -28 57 67 21 -32 66 32 44 47 45 28 -52 83 -37 74 -86 99 -69 33 -81 5 43 7 16 -34 38 -46 -62 -57 21 93 -41 -8 -3 -86 83 7 41 15 -31 12 -91 75 -51 -7 56 86 51 90 60 82 -57 50 15 77 28 -51 -49 -46 30 11 -7 0 -74 51 -32 56 -14 -51 91 6 -43 15 -36 6 -5 68 -42 -78 10 51 2 -52 -17 38 50 9 77 19 -15 33 -44 -14 85 57 -81 9 49 -3 -76 -100 0 2 99 -46 24 27 -14 -92 -19 60 90 -72 44 78 -91 -55 -83 -13 -99 58 79 66 63 -42 -50 70 30 6 -14 86 -82 61 98 9 -49 99 -51 -71 -86 -78 -77 68 -62 24 42 -73 42 -93 47 73 -9 -38 -4 -35 -79 33 39 41 -97 53 97 79 21 -5 42 -71 -71 -66 26 -90 96 87 -28 78 35 -89 28 -62 15 -78 -72 22 58 13 -6 55 -36 32 -12 17 -40 8 -75 -39 -75 -57 -54 42 82 12 97 8 -20 37 77 4 90 18 30 -90 -95 0 -75 81 25 -26 -79 -92 -100 86 65 -3 45 -91 49 -13 62 -89 52 -9 64 85 -44 -98 -76 20 -59 -80 35 90 65 46 -98 -22 12 59 -18 -56 35 50 -78 -63 42 96 47 0 62 -29 87 -11 41 78 65 -62 -55 -81 -56 -78 57 -95 -67 -100 -28 -91 -80 -81 -6 -33 -3 90 -95 18 -25 -98 -32 27 -58 -41 -45 23 58 70 -68 -54 -52 86 80 63 -67 -8 14 -41 -13 30 17 -78 64 33 75 28 -94 -6 74 97 -16 -77 3 63 70 17 76

відсортовано: -100 -100 -100 -100 -100 -100 -99 -99 -98 -98 -98 -98 -98 -98 -98 -98 -97 -97 -97 -97 -97 -97 -96 -96 -96 -96 -96 -96 -95 -95 -95 -95 -94 -94 -94 -94 -93 -93 -93 -93 -92 -92 -92 -91 -91 -91 -91 -91 -91 -91 -91 -90 -90 -90 -90 -89 -89 -89 -89 -89 -89 -88 -88 -87 -87 -86 -86 -86 -86 -86 -86 -86 -85 -85 -85 -84 -83 -83 -83 -82 -82 -82 -82 -81 -81 -81 -81 -81 -81 -81 -81 -80 -80 -80 -79 -79 -79 -79 -79 -79 -79 -78 -78 -78 -78 -78 -78 -78 -78 -77 -77 -77 -77 -77 -77 -76 -76 -76 -76 -76 -76 -75 -75 -75 -75 -75 -75 -75 -75 -75 -74 -74 -74 -74 -73 -73 -73 -73 -73 -73 -72 -72 -72 -72 -72 -72 -71 -71 -71 -71 -71 -71 -71 -71 -70 -70 -69 -69 -69 -69 -69 -69 -68 -68 -68 -68 -67 -67 -67 -67 -67 -67 -66 -66 -66 -65 -65 -65 -64 -64 -64 -64 -63 -63 -63 -62 -62 -62 -62 -62 -62 -62 -62 -61 -61 -61 -60 -59 -59 -59 -58 -58 -58 -57 -57 -57 -57 -57 -57 -56 -56 -56 -56 -56 -56 -56 -55 -55 -55 -55 -55 -54 -54 -54 -54 -53 -53 -52 -52 -52 -52 -52 -51 -51 -51 -51 -51 -51 -51 -51 -51 -50 -50 -49 -49 -49 -49 -49 -49 -48 -48 -48 -48 -47 -47 -47 -46 -46 -46 -46 -45 -45 -45 -44 -44 -44 -44 -43 -43 -43 -42 -42 -42 -42 -42 -42 -42 -42 -42 -41 -41 -41 -41 -41 -41 -40 -40 -40 -40 -39 -39 -39 -39 -38 -38 -38 -38 -37 -37 -37 -36 -36 -36 -36 -36 -36 -35 -35 -35 -34 -34 -34 -34 -34 -33 -33 -33 -33 -33 -32 -32 -32 -32 -32 -32 -32 -31 -31 -31 -31 -31 -31 -31 -31 -30 -30 -30 -29 -29 -29 -29 -29 -29 -28 -28 -28 -28 -28 -28 -27 -27 -27 -27 -27 -27 -27 -27 -27 -26 -26 -26 -26 -25 -25 -24 -24 -24 -23 -23 -22 -22 -21 -21 -21 -21 -21 -20 -20 -20 -20 -20 -19 -19 -19 -18 -18 -17 -17 -17 -17 -17 -17 -17 -17 -17 -16 -16 -16 -16 -16 -16 -16 -16 -16 -15 -15 -15 -14 -14 -14 -14 -14 -14 -14 -14 -14 -14 -14 -13 -13 -13 -13 -13 -13 -13 -12 -12 -12 -11 -11 -11 -11 -11 -11 -10 -10 -10 -10 -9 -9 -9 -9 -9 -9 -8 -8 -8 -8 -8 -7 -7 -7 -7 -7 -7 -6 -6 -6 -6 -6 -6 -6 -5 -5 -5 -5 -4 -4 -4 -4 -3 -3 -3 -3 -3 -3 -3 -2 -1 -1 0 0 0 0 0 0 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 4 4 4 4 5 5 5 6 6 6 6 6 6 7 7 7 7 7 8 8 8 8 8 8 9 9 9 9 9 9 10 11 11 12 12 12 12 12 13 13 13 13 13 13 14 14 14 14 14 14 14 15 15 15 15 15 15 15 16 16 16 16 16 17 17 17 17 17 17 17 17 17 18 18 18 18 18 19 19 20 20 20 21 21 21 21 21 22 22 22 22 22 22 22 22 22 23 23 24 24 24 25 25 25 25 25 25 26 26 26 27 27 27 27 27 28 28 28 28 28 28 28 29 29 29 30 30 30 30 30 30 31 31 31 31 31 32 32 32 32 32 32 33 33 33 33 33 34 34 34 34 34 35 35 35 35 35 35 35 35 35 35 36 36 36 37 37 37 37 37 37 38 38 38 38 39 39 39 39 39 40 41 41 41 41 41 41 41 42 42 42 42 42 42 42 42 43 43 43 43 43 43 43 44 44 44 44 44 45 45 45 45 46 46 46 46 47 47 47 47 47 47 48 49 49 49 49 49 49 50 50 50 50 50 50 51 51 51 51 51 51 51 52 52 52 52 53 53 54 54 54 54 55 55 55 55 56 56 56 56 57 57 57 57 57 57 57 58 58 58 58 58 58 59 59 59 59 59 60 60 60 60 60 60 60 60 61 61 61 61 62 62 62 62 62 63 63 63 63 63 63 63 63 63 64 64 64 64 65 65 65 65 65 65 65 66 66 66 66 66 66 66 66 66 67 67 68 68 68 69 69 69 70 70 70 70 70 70 70 70 70 70 71 71 71 71 71 72 72 72 73 73 73 73 74 74 74 74 74 74 75 75 75 75 76 76 76 76 76 76 77 77 77 77 77 77 78 78 78 78 78 78 78 78 79 79 79 79 79 79 80 80 80 80 80 80 80 81 81 81 81 81 81 81 82 82 82 82 82 82 82 83 83 83 83 84 84 85 85 85 85 85 85 85 85 86 86 86 86 86 86 87 87 87 87 87 88 88 88 88 89 89 89 90 90 90 90 90 90 91 91 92 92 92 92 93 93 93 93 93 94 94 94 95 95 95 95 96 96 96 96 96 96 96 96 96 96 97 97 97 97 98 98 98 98 98 99 99 99 99 99 99 99 99 99

* Результати тестування програми:
  + - * 3 елемента:
      * 9 елементів:
      * 100 елементів:
      * 1000 елементів:
* Аналіз отриманих результатів:

В ході тестування було визначено час роботи алгоритму сортування злиттям за допомогою Stopwatch(). Для чистоти експерименту тестування проводилось на однаковій послідовності елементів масиву та списку. Переглянувши отримані результати і проаналізувавши алгоритми сортування масиву і списку, можна зробити висновок, що сортування злиттям на цих структурах потребує приблизно однаковий час. Це пов’язано з тим, що при сортуванні масиву для поділу масиву на 2 частини використовується перезапис окремих частин масиву у масиви LeftArr і RightArr, в той час як при сортуванні списку також за допомогою циклу перебираються всі елементи з метою пошуку середнього елементу. Також можна помітити, що різниця у часі не залежить від об'єму вхідних даних.

* Висновок:

В ході цієї лабораторної роботи було розроблено алгоритм сортування злиттям на масиві і списку. Було розроблено набір тестів і порівняно час роботи на різних структурах даних: масиві і лінійному списку. Різниця у часі є досить незначною і не залежить від об’єму вхідних даних.