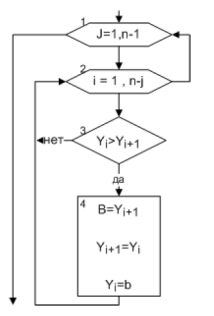
Отчет по лабе 2  
Гомзякова Наталья

Сортировка Пузырьком

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Время | 100 элемент | 1000 Элемент | 10000 Элемент | 10000 Элемент |
| 1 | 0 | 0,02 | 0,97 | 00:01:31.20 |
| 2 | 0 | 0,01 | 1,07 | 00:01:31.83 |
| 3 | 0 | 0,01 | 1,01 | 00:01:31.37 |
| 4 | 0 | 0,02 | 1,09 | 00:01:31.74 |
| 5 | 0 | 0,01 | 1,34 | 00:01:31.89 |
| 6 | 0 | 0,02 | 1,06 | 00:01:31.93 |
| 7 | 0 | 0,01 | 1,12 | 00:01:31.78 |
| 8 | 0 | 0,01 | 1,05 | 00:01:31.81 |
| 9 | 0 | 0,01 | 1,12 | 00:01:31.96 |
| 10 | 0 | 0,01 | 1,01 | 00:01:31.43 |
| Сред | 0 | 0,013 | 1,084 | 00:01:31.69 |

Код реализации:

static int[] BubbleSort(ref int[] bst)

{

int temp;

for (int i = 0; i < bst.Length; i++)

{

for (int j = i + 1; j < bst.Length; j++)

{

if (bst[i] > bst[j])

{

temp = bst[i];

bst[i] = bst[j];

bst[j] = temp;

}

}

}

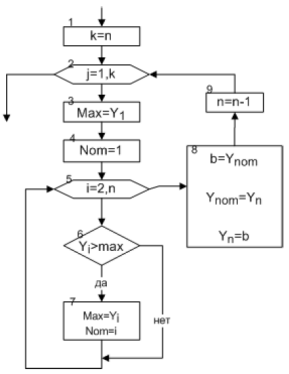
return bst;

}

Сортировка Выбором

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Время | 100 элемент | 1000 Элемент | 10000 Элемент | 10000 Элемент |
| 1 | 0 | 0,1 | 0,54 | 47,94 |
| 2 | 0 | 0,1 | 0,57 | 48,4 |
| 3 | 0 | 0,1 | 0,54 | 48,08 |
| 4 | 0 | 0,1 | 0,52 | 48,12 |
| 5 | 0 | 0,1 | 0,55 | 48,08 |
| 6 | 0 | 0,1 | 0,55 | 48,12 |
| 7 | 0 | 0,1 | 0,56 | 48,37 |
| 8 | 0 | 0,1 | 0,62 | 48,23 |
| 9 | 0 | 0,1 | 0,56 | 48,47 |
| 10 | 0 | 0,1 | 0,55 | 48,08 |
| Сред | 0 | 0,1 | 0,556 | 48,189 |

Код реализации

static int[] ViborSort(ref int[] bst)

{

for (int i = 0; i < bst.Length - 1; i++)

{

//поиск минимального числа

int min = i;

for (int j = i + 1; j < bst.Length; j++)

{

if (bst[j] < bst[min])

{

min = j;

}

}

//обмен элементов

int temp = bst[min];

bst[min] = bst[i];

bst[i] = temp;

}

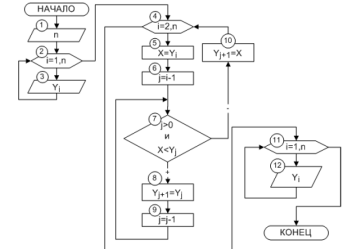
return bst;

}

Сортировка вставка

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Время | 100 элемент | 1000 Элемент | 10000 Элемент | 10000 Элемент |
| 1 | 0 | 0 | 0,35 | 33,74 |
| 2 | 0 | 0 | 0,39 | 33,44 |
| 3 | 0 | 0 | 0,4 | 33,01 |
| 4 | 0 | 0,1 | 0,42 | 33,33 |
| 5 | 0 | 0 | 0,4 | 34 |
| 6 | 0 | 0,1 | 0,4 | 34,23 |
| 7 | 0 | 0 | 0,51 | 33,73 |
| 8 | 0 | 0 | 0,41 | 33,92 |
| 9 | 0 | 0 | 0,44 | 33,81 |
| 10 | 0 | 0,1 | 0,42 | 34,78 |
| Сред | 0 | 0,03 | 0,414 | 33,799 |

Код реализации:

static int[] InsertionSort( ref int[] bst)

{

int newElement, location;

for (int i = 1; i < bst.Length; i++)

{

newElement = bst[i];

location = i - 1;

while (location >= 0 &&

bst[location] > newElement)

{

bst[location + 1] = bst[location];

location = location - 1;

}

bst[location + 1] = newElement;

}

return bst;

}

Вывод:

|  |  |  |  |
| --- | --- | --- | --- |
| Сортировка | Пузырек | Выборка | Вставка |
| 100 | 00:00:00.00 | 00:00:00.00 | 00:00:00.00 |
| 1000 | 00:00:00.01 | 00:00:00.01 | 00:00:00.03 |
| 10000 | 00:00:01.87 | 00:00:00.55 | 00:00:00.41 |
| 100000 | 00:01:31.69 | 00:00:48.18 | 00:00:33.79 |

Сортировка Вставкой является очень эффективной, тк занимает меньше времени на обработку