Для начала глянем на код метода ***Reverse():***

```
public static void Reverse(Array array)
{
  if (array == null)
    throw new ArgumentNullException(nameof (array));
  Array.Reverse(array, array.GetLowerBound(0), array.Length);
}
```

```
[SecuritySafeCritical]
[ReliabilityContract(Consistency.MayCorruptInstance, Cer.MayFail)]
[__DynamicallyInvokable]
public static void Reverse(Array array, int index, int length)
{
  if (array == null)
    throw new ArgumentNullException(nameof (array));
  if (index < array.GetLowerBound(0) || length < 0)
    throw new ArgumentOutOfRangeException(index < 0 ? nameof (index) : nameof (length), Environment.GetResourceString("Argumer
  if (array.Length - (index - array.GetLowerBound(0)) < length)
    throw new ArgumentException(Environment.GetResourceString("Argument_InvalidOffLen"));
  if (array.Rank != 1)
    throw new RankException(Environment.GetResourceString("Rank_MultiDimNotSupported"));
  if (Array.TrySZReverse(array, index, length))
    return;
  int index1 = index;
  int index2 = index + length - 1;
  if (array is object[] objArray)
  {
    for (; index1 < index2; --index2)
    {
      object obj = objArray[index1];
      objArray[index1] = objArray[index2];
      objArray[index2] = obj;
      ++index1;
    }
  }
  else
  {
    for (; index1 < index2; --index2)
    {
      object obj = array.GetValue(index1);
      array.SetValue(array.GetValue(index2), index1);
      array.SetValue(obj, index2);
      ++index1;
    }
  }
}
```

Можно заметить, что код в красном угольнике ***совпадает*** с реализованным в задании HW_05_03.
Однако в данном методе он используется ***только для ссылочных*** типов данных.
При использовании массивов **значимых** типов Visual Studio ***выдаёт ошибку*** :
CS8121: Expression of type T cannot be handled by a pattern of type X
Поэтому для значимых типов используется код в оранжевом угольнике.
Сравнивая результаты приведения массива типа int методом ***Reverse()*** и написанного «ручками» получаем преимущество в производительность примерно ***в 5 раз*** в пользу метода ***Reverse().***

Как одну из причин почему значимые типы не используют код в красном угольнике, я вижу операции ***boxing/ unboxing***, которые требуют большое количество ресурсов, что и стало заметно при массиве в 100 миллионов элементов.
Другие причины хотелось бы узнать ☺
Вадим, если есть свободная минутка, расскажите, пожалуйста ☺.