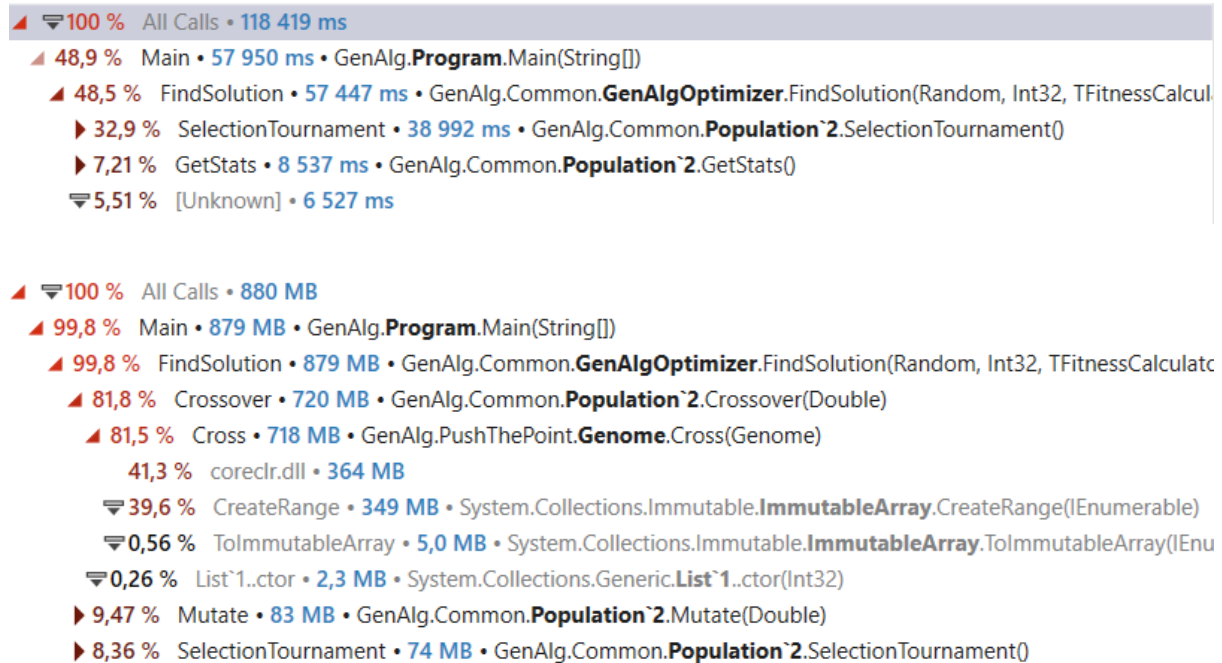
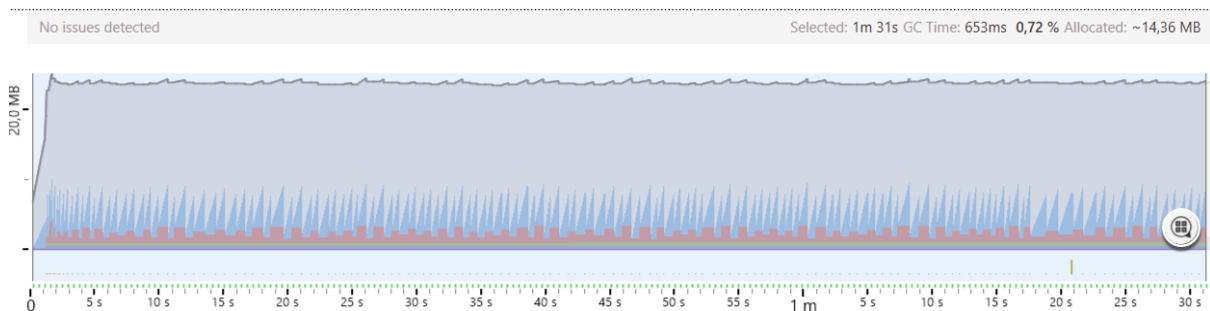


# Initial Implementation

Первичная реализация генетического алгоритма



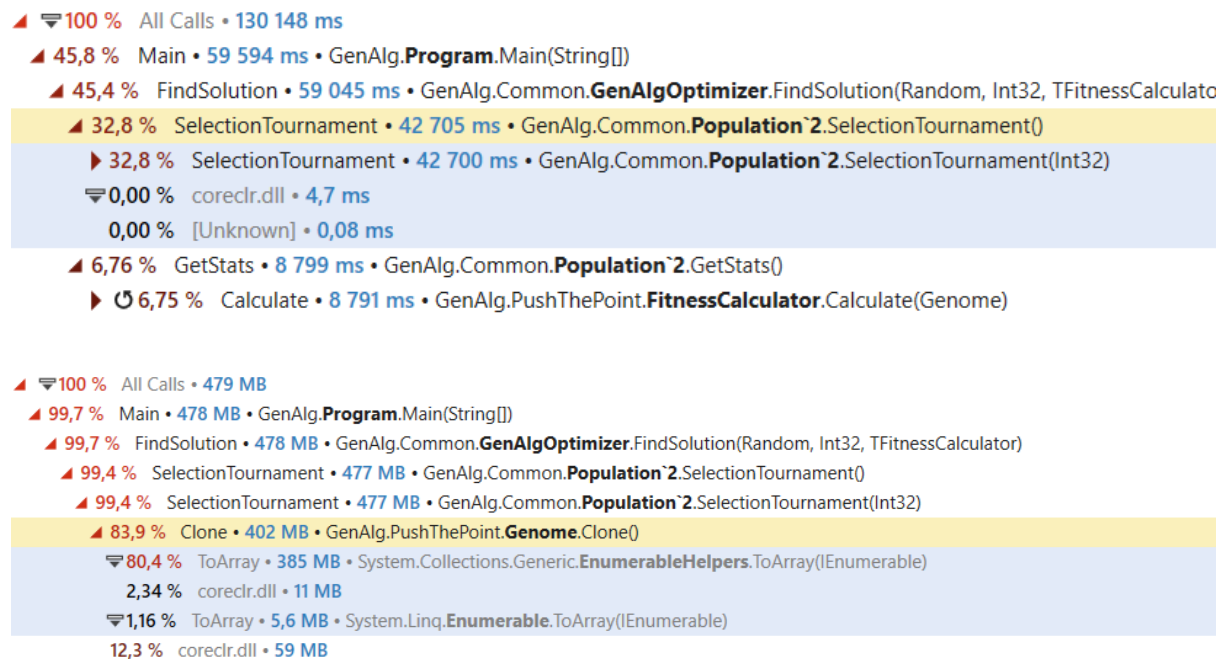
Profiling session has finished



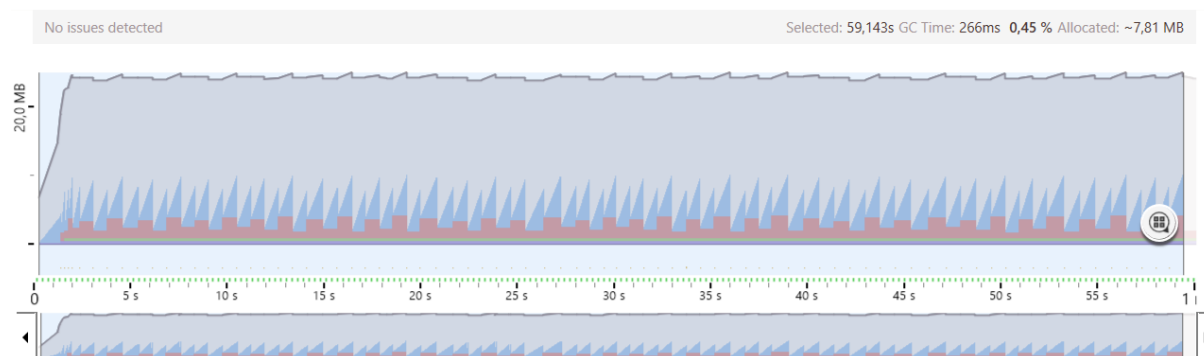
# ImmutableArray => array[]

Геном перестал быть неизменяемым объектом и операции мутации и скрещивания делаются без дополнительного выделения памяти.

Выделение памяти снизилось с 880 мб до 479 мб.



Profiling session has finished

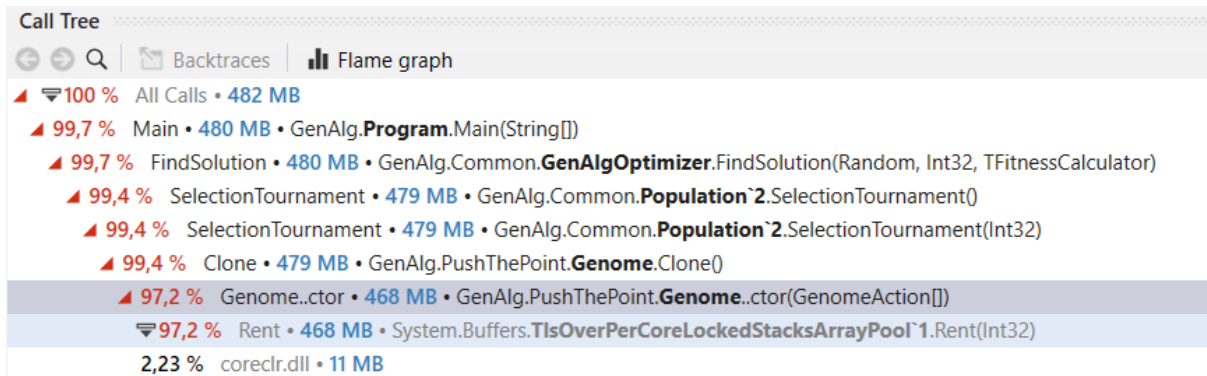


# Array Pool

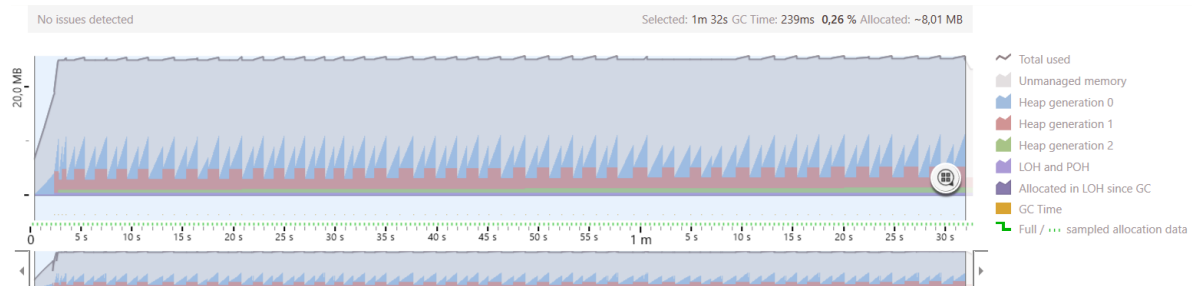
Вместо выделения памяти для новых массивов вместо операции `new []` был использован `ArrayPool<>.Shared.Rent`.

Выделение памяти не изменилось.

```
▲ 100 % All Calls • 130 044 ms
▲ 49,3 % Main • 64 112 ms • GenAlg.Program.Main(String[])
▲ 49,0 % FindSolution • 63 712 ms • GenAlg.Common.GenAlgOptimizer.FindSolution(Random, Int32, TFitnessCalculator)
  ▶ 34,7 % SelectionTournament • 45 069 ms • GenAlg.Common.Population`2.SelectionTournament()
  ▶ 8,73 % GetStats • 11 354 ms • GenAlg.Common.Population`2.GetStats()
```



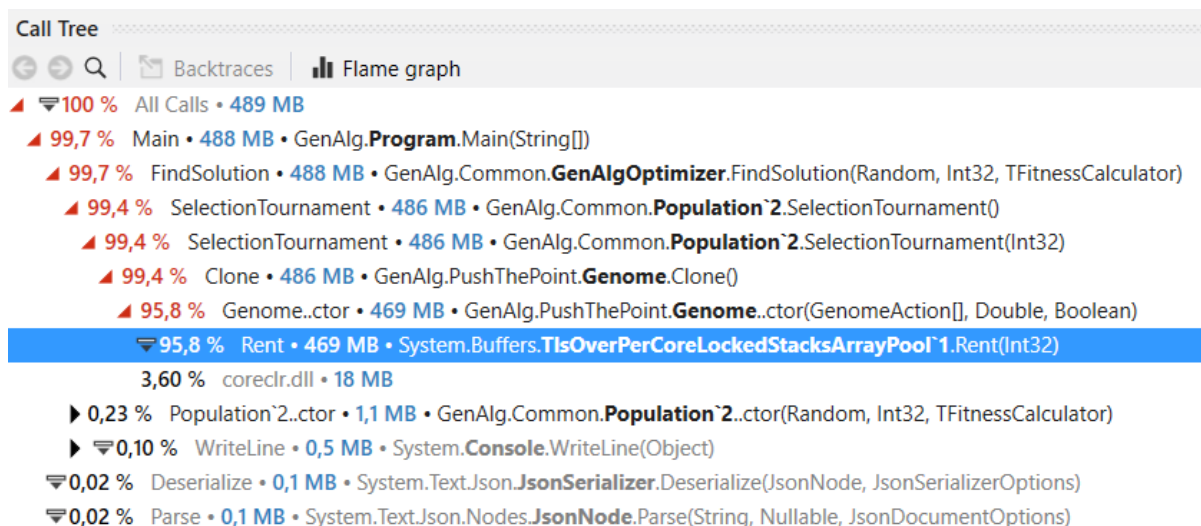
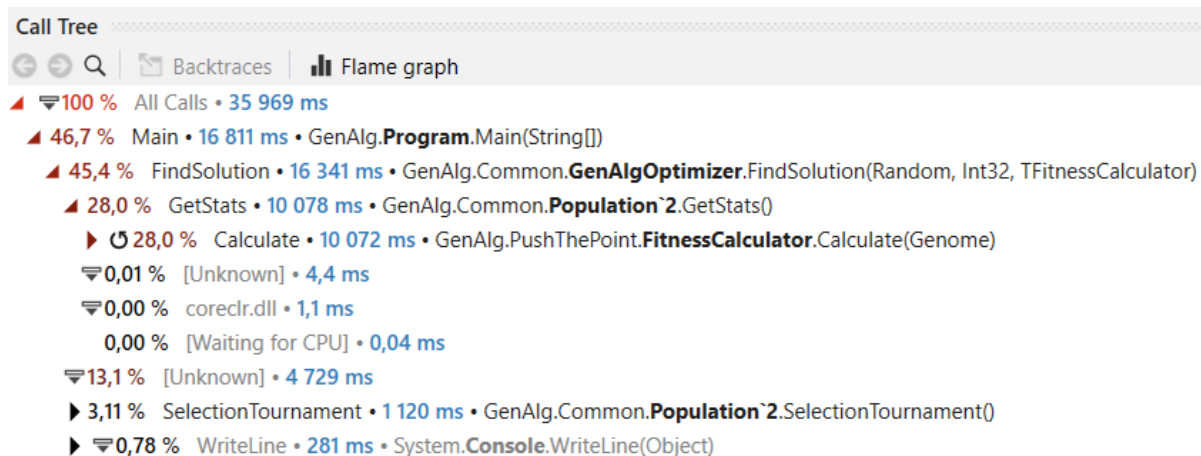
Profiling session has finished



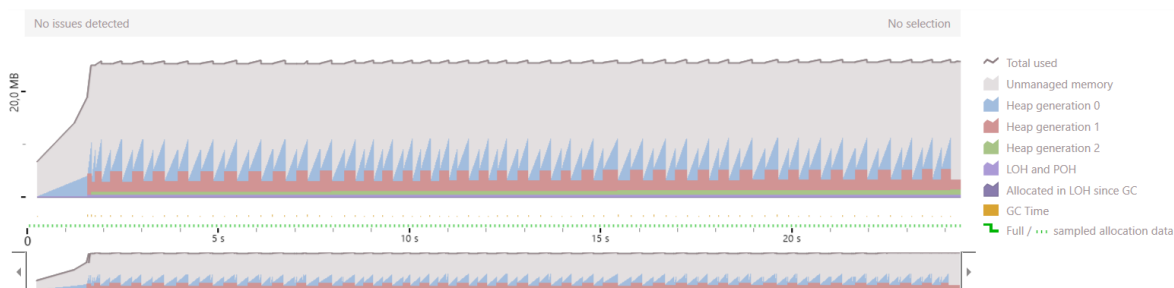
# Cache Fitness

Операция определения fitness представляет собой алгоритм, так что она достаточно дорогая. Теперь для неизмененного генома fitness вычисляется 1 раз и кэшируется.

Скорость работы выросла в 3.4 раза.



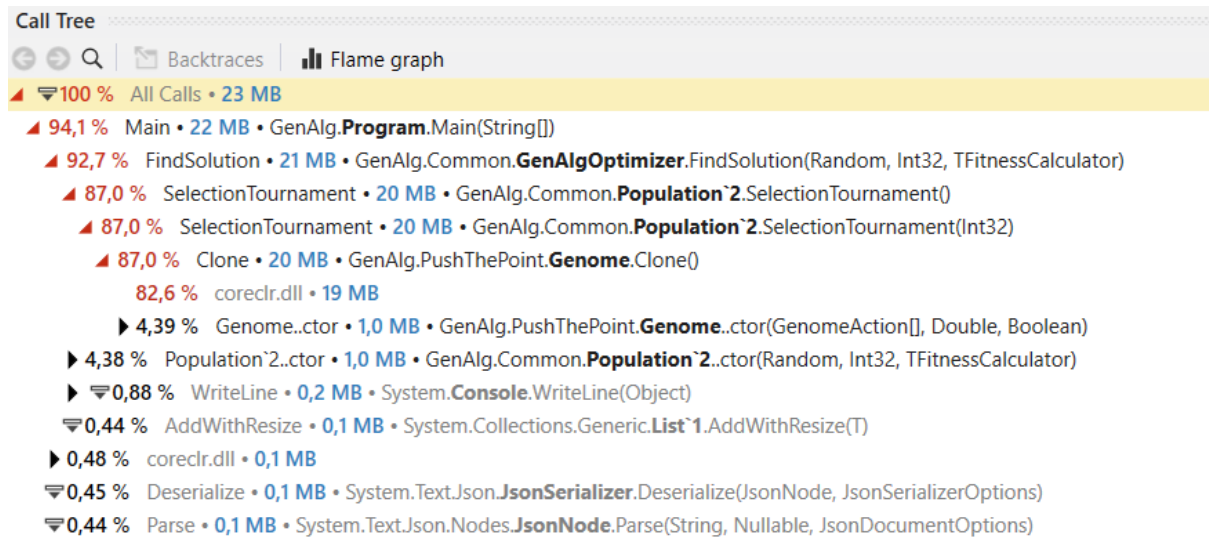
Profiling session has finished



## ArrayPool.Create

Теперь вместо использования общего `ArrayPool` создаётся отдельный объект методом `ArrayPool.Create` и в дальнейшем используется для выделения памяти для массивов.

Выделение памяти снизилось с 480 мб до 23 мб.



## Итог

Удалось снизить время работы программы со 118 секунд до 36 секунд.

Удалось снизить выделение памяти с 880 мб до 23 мб.