

Word Search

Aufgabe

- Eine Wortliste mit zufälliger Reihenfolge
- Kombinationen von 4 Großbuchstaben bestehend aus AAAA bis ZZZZ
- Alle Wörter, deren erstes Zeichen mit einem Suchstring übereinstimmen, sollen gefunden werden (duplicates!)

Demo

Erstellen der Wortliste

- Auch das kann schon optimiert werden
 - Verwendung von `StringSpan`
 - `StringCompareOrdinal`
- Erstellen großer Wortliste (Dim>4) sehr zeitintensiv

WordGenerator Benchmarks

BenchmarkDotNet v0.13.12, Windows 10 (10.0.19045.4291/22H2/2022Update)

AMD Ryzen 5 3600, 1 CPU, 12 logical and 6 physical cores

.NET SDK 7.0.402

[Host] : .NET 7.0.12 (7.0.1223.47720), X64 RyuJIT AVX2

DefaultJob : .NET 7.0.12 (7.0.1223.47720), X64 RyuJIT AVX2

Method	Length	Mean	Error	StdDev
GenerateRecursiveStringSpan	4	33.750 ms	0.4565 ms	0.4270 ms
GenerateFixedSizeString	4	14.833 ms	0.1358 ms	0.1204 ms
GenerateFixedSizeStringSpan	4	6.617 ms	0.0555 ms	0.0519 ms
GenerateDynamicStringSpan	4	33.762 ms	0.3318 ms	0.3104 ms
GenerateRecursiveStringSpan	5	740.602 ms	14.6072 ms	31.7548 ms
GenerateFixedSizeString	5	718.821 ms	10.2870 ms	9.1192 ms
GenerateFixedSizeStringSpan	5	146.162 ms	1.3429 ms	1.1905 ms
GenerateDynamicStringSpan	5	757.661 ms	15.0094 ms	27.8209 ms

Linear Search

- geeignet für unsortierte Listen
- gut zu parallelisieren (Jeder Thread bekommt einen Teil der Liste)
- Time Complexity ist $O(n)$

Linear Search Benchmarks

BenchmarkDotNet v0.13.12, Windows 10 (10.0.19045.4291/22H2/2022Update)

AMD Ryzen 5 3600, 1 CPU, 12 logical and 6 physical cores

.NET SDK 7.0.402

[Host] : .NET 7.0.12 (7.0.1223.47720), X64 RyuJIT AVX2

DefaultJob : .NET 7.0.12 (7.0.1223.47720), X64 RyuJIT AVX2

Method	dimensions	Mean	Error	StdDev
LinearSearch	4	5.128 ms	0.1008 ms	0.1035 ms
ParallelLinearSearch	4	1.008 ms	0.0162 ms	0.0136 ms
ParallelLinearSearchConcurrentQueue	4	1.190 ms	0.0235 ms	0.0261 ms
LinearSearch	5	269.339 ms	1.2448 ms	1.0395 ms
ParallelLinearSearch	5	105.000 ms	0.4784 ms	0.4475 ms
ParallelLinearSearchConcurrentQueue	5	106.263 ms	0.2682 ms	0.2378 ms

Binary Search

- nur für sortierte Listen
- Time Complexity ist $O(\log n)$
- eignet sich für sehr große Datensätze

Binary Search Benchmarks



BenchmarkDotNet v0.13.12, Windows 10 (10.0.19045.4291/22H2/2022Update)

AMD Ryzen 5 3600, 1 CPU, 12 logical and 6 physical cores

.NET SDK 7.0.402

[Host] : .NET 7.0.12 (7.0.1223.47720), X64 RyuJIT AVX2

DefaultJob : .NET 7.0.12 (7.0.1223.47720), X64 RyuJIT AVX2

Method	Dimension	Mean	Error	StdDev
ParallelLinearSearch	4	1,057.29 µs	19.332 µs	37.246 µs
BinarySearch	4	25.29 µs	0.125 µs	0.111 µs
Sort	4	 507,378.23 µs	9,475.524 µs	8,863.411 µs
ParallelLinearSearch	5	104,252.74 µs	313.481 µs	293.230 µs
BinarySearch	5	33.37 µs	0.270 µs	0.226 µs
Sort	5	 30,141,129.81 µs	59,446.513 µs	49,640.531 µs