

Softwarewettbewerb

Auswertung

Die Teams und ihre Implementierungen

- Team 1: TrigramStorageImplementation
- Team 2: ThreeGramStorage
- Team 3: ComplexTrigramStorage
- Team 4: OMGOptimizedTrigramStorage
- Team 5: TrieTrigramStorage
- Team 6: TreegramStorage
- Team 7: TreeTrigramStorage
- Team 8: Treap
- Team 9: MyTrigramStorage
- Team 10: AdvancedTrigramStorage
- Team 11: AVLPriorityHeapTrigramStorage
- Team 12: FastTrigramStorage
- Team 13: WeirdTrigramStorage
- Team 14: SimpleTrigramStorage

Die Teams und ihre Implementierungen

- Team 1: TrigramStorageImplementation
- Team 2: ThreeGramStorage
- Team 3: ComplexTrigramStorage
- Team 4: OMGOptimizedTrigramStorage
- Team 5: TrieTrigramStorage
- Team 6: TreegramStorage
- Team 7: TreeTrigramStorage
- Team 8: Treap
- Team 9: MyTrigramStorage
- Team 10: AdvancedTrigramStorage
- Team 11: AVLPriorityHeapTrigramStorage
- Team 12: FastTrigramStorage
- Team 13: WeirdTrigramStorage
- Team 14: SimpleTrigramStorage
- **Team 15: wir (mwahahaha!)**

Probleme

Sortiert speichern?

Wie Positionen finden?

Wie Summe bestimmen?

Wie Maximum bestimmen?

Lösungsansätze

Sortieren

- QuickSort, Heapsort, etc.
- Wörter \rightarrow int

Positionen finden

- Binärsuche

Summe bestimmen

- laufende Summe im Voraus berechnen
- die Summe eines Bereichs ist Differenz laufenden Summen an den Endpunkten

Wer wie?

- Team 1: HeapSort, Binärsuche, lineare Suche (≈ 170 LOC)
- Team 2: ? ? ? (Schachtelungstiefe 8) (≈ 1050 LOC)
- Team 3: AVL Baum, Hashing (≈ 630 LOC)
- Team 4: QuickSort, “Ranking”, binärer Baum (≈ 260 LOC)
- Team 5: Trie, laufende Summe, Skiplist für Maximum (≈ 940 LOC)
- Team 6: QuickSort, balancierter Baum, laufende Summe, Skiplist für Max (≈ 250 LOC)
- Team 7: annotierter 37-ärer Baum (Trie?) (≈ 180 LOC)
- Team 8: Treap (count als prio), laufende Summe (≈ 330 LOC)
- Team 9: QuickSort, laufende Summe, Skiplist für Maximum (≈ 210 LOC)
- Team 10: Trie, lineare Suche (≈ 250 LOC)
- Team 11: HashMap, PriorityHeap, laufende Summe (≈ 640 LOC)
- Team 12: Hashing, laufende Summe, ? (≈ 360 LOC)
- Team 13: HeapSort, laufende Summe, Treap, Hashing, Trie (≈ 920 LOC)
- Team 14: Trie, Skiplist?, laufende Summe (≈ 1800 LOC)
- Team 15: QuickSort, lauf. Summe, Skiplist für Max, Int-Vokabular (≈ 300 LOC)

Schnipsel I

```
//n ist Blatt -> wir haben gefunden, was wir suchten
if(n instanceof TrieLeaf){

    if(stack.isEmpty()){
        //Stack ist leer, wir koennen nicht zurueck

// force the JIT compiler to do its work
JITOptimizer.optimize(this, startTime, 19, 1000000);

/*
Autoren:
Daniel Mock
Joshua Fuerste
Daniel Alexander Mock
```

Schnipsel II

```
if (u == this.root) { //ACHTUNG r
    this.root = w;
```

```
DoubleBinMaxHeapTreeNode(TrigramAndCount tc) {
```

```
public void insert ( int key, long prio ) {
    TreapNode n = new TreapNode (key, prio );
    bstIns (n);
    bubble (n);
}
```

```
public class TrigramAndCountTree2LeonEdition2 {
```


Testdaten: Wikipedia-Dump

- Preprocessing:
 - lower case only
 - Löschen ungültiger verbleibender Zeichen
 - nur Trigramme, die mind. 2 Mal vorkommen
 - nur Trigramme mit häufigsten Wörtern
 - ...
- 93754779 Trigramme
- 100 “Warmup”-Queries
- 500k Queries (je 100k klein, mittel, groß, sehr groß und “unbekannt”)
- Trigramme und Queries bald verfügbar (Speicherverbrauch ca. 60 bis 180GB)
- Implementierungen (teilweise) und Folien ebenso

Alle dabei?

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Alle dabei?

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Große Daten fordern Opfer . . .

2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 15

Alle dabei?

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Große Daten fordern Opfer . . .

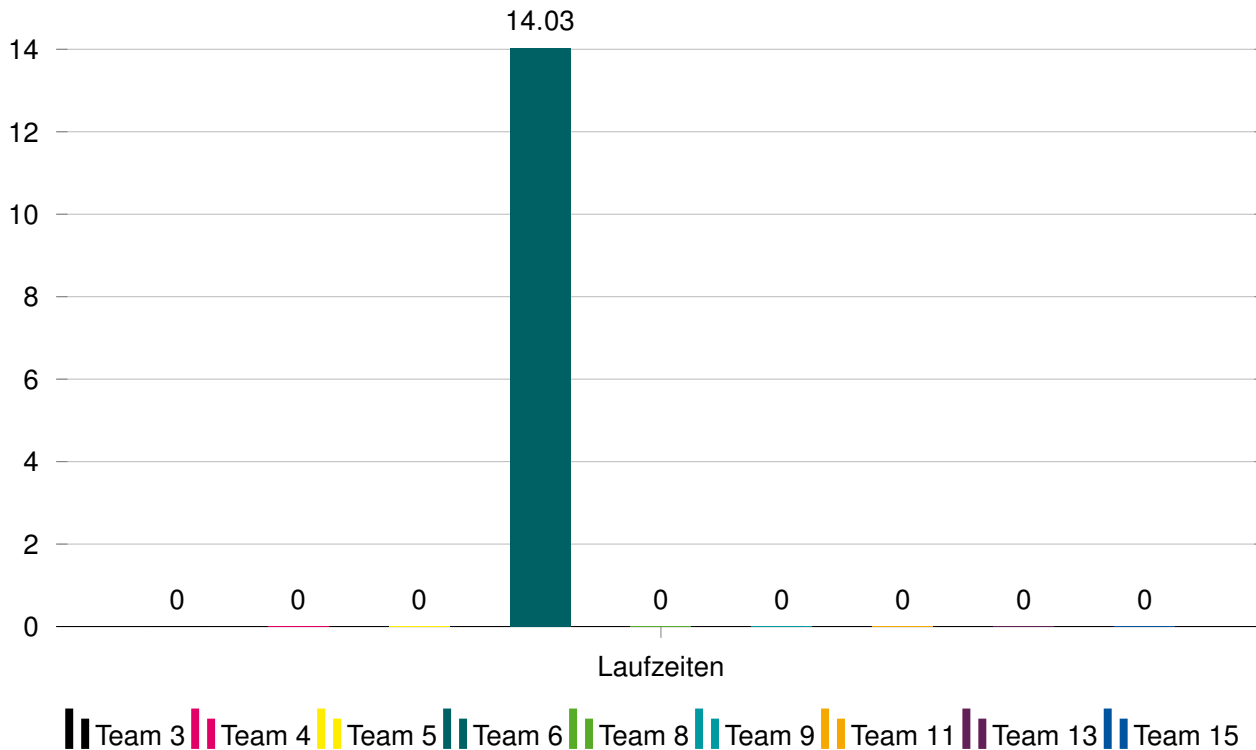
2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 15

Alles eine Frage der Korrektheit . . .

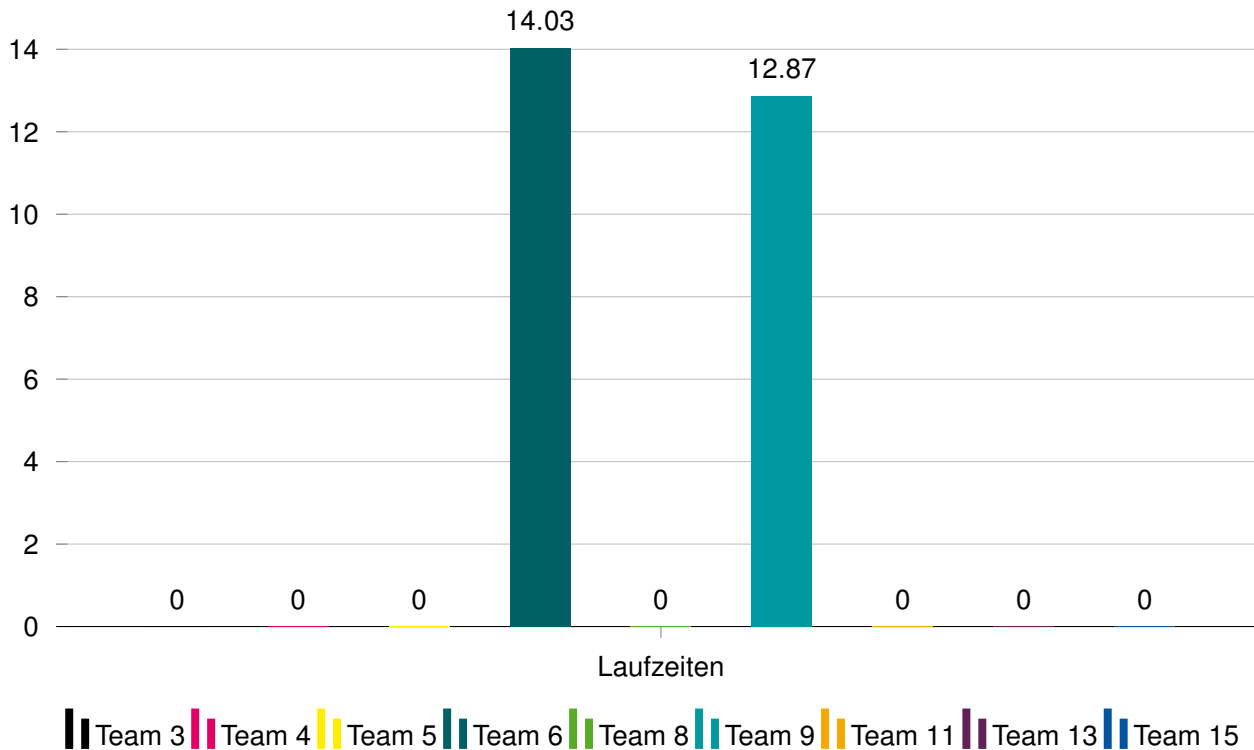
3, 4, 5, 6, 8, 9, 11, 13, 15

Siegerehrung

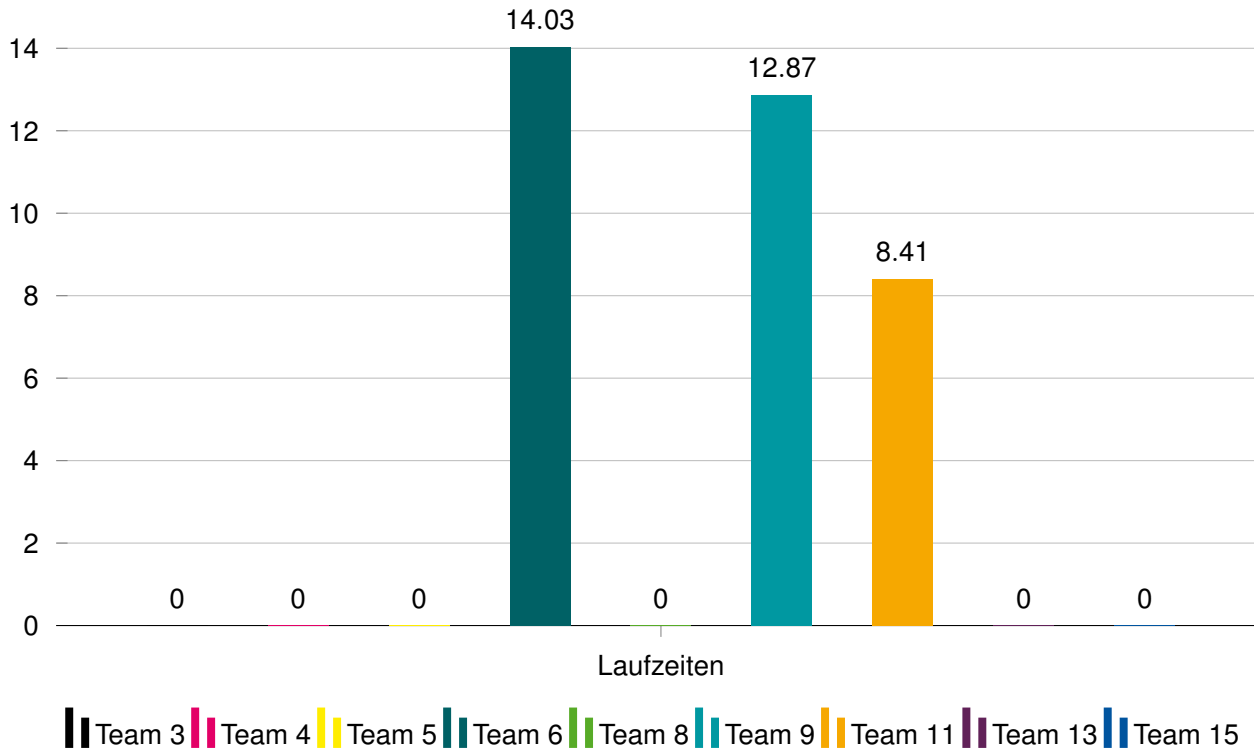
Siegerehrung



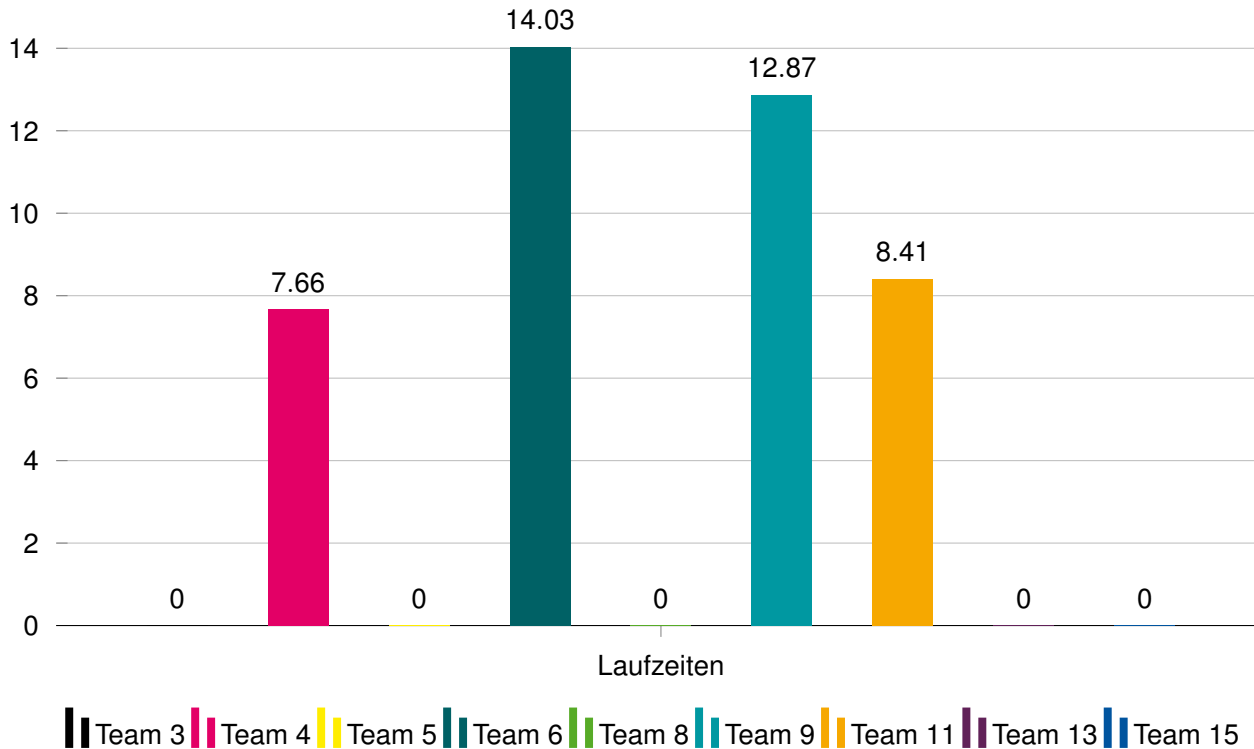
Siegerehrung



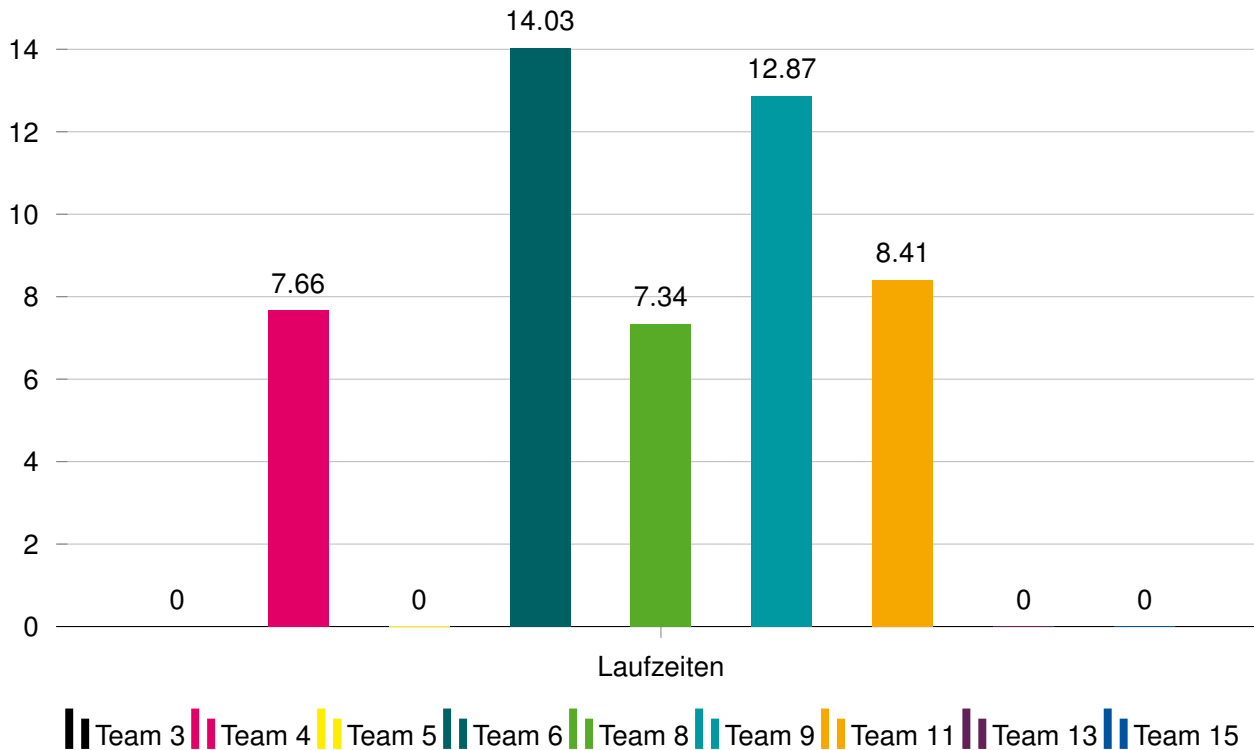
Siegerehrung



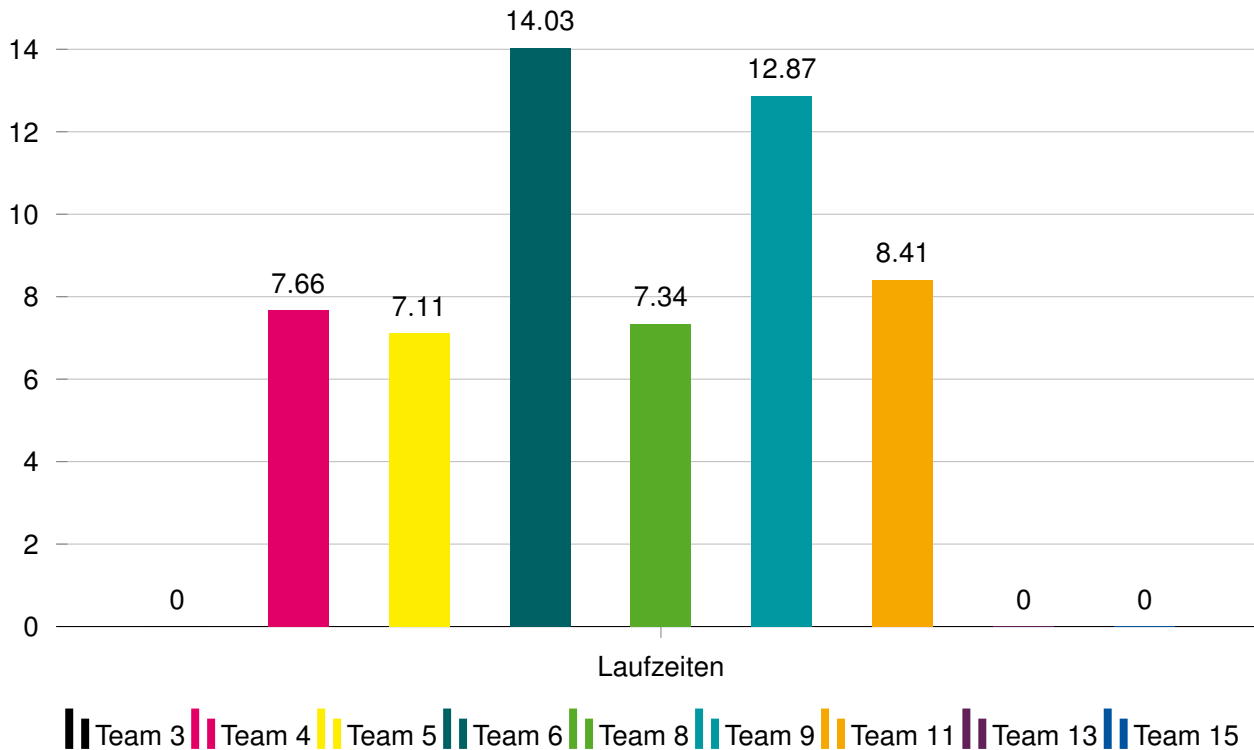
Siegerehrung



Siegerehrung

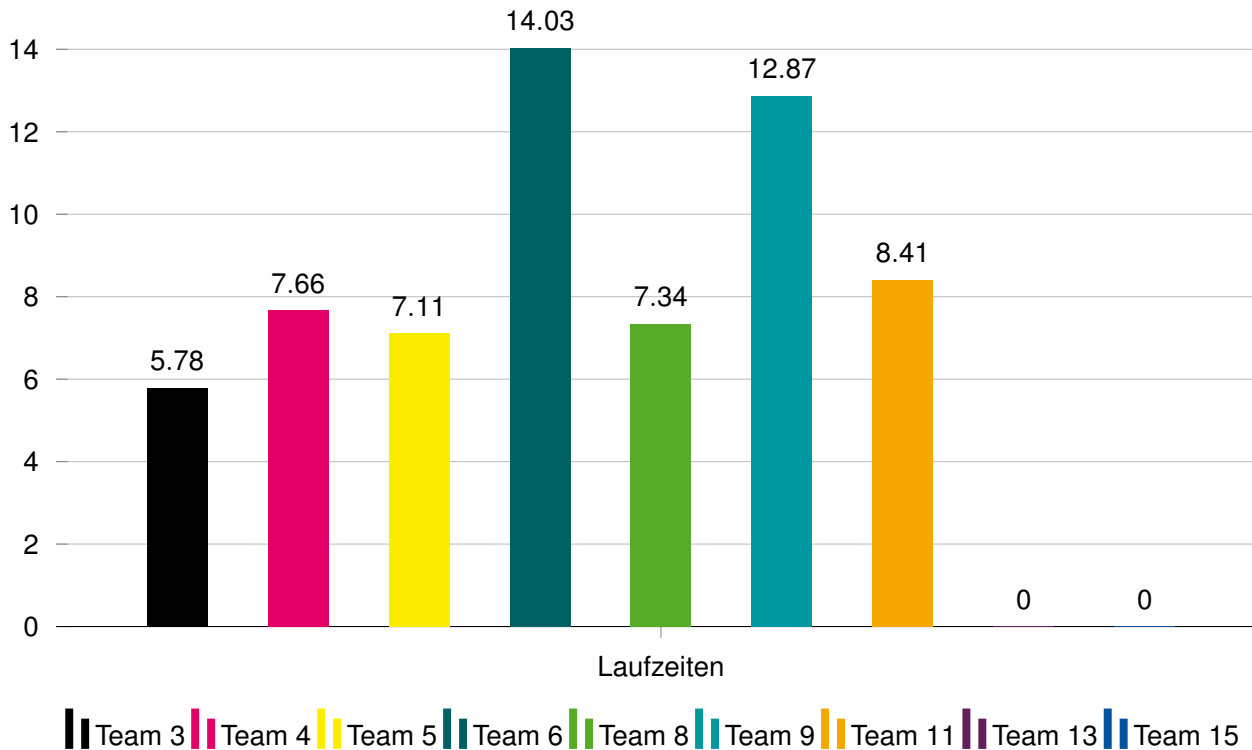


Siegerehrung



Platz 3:
Team 5

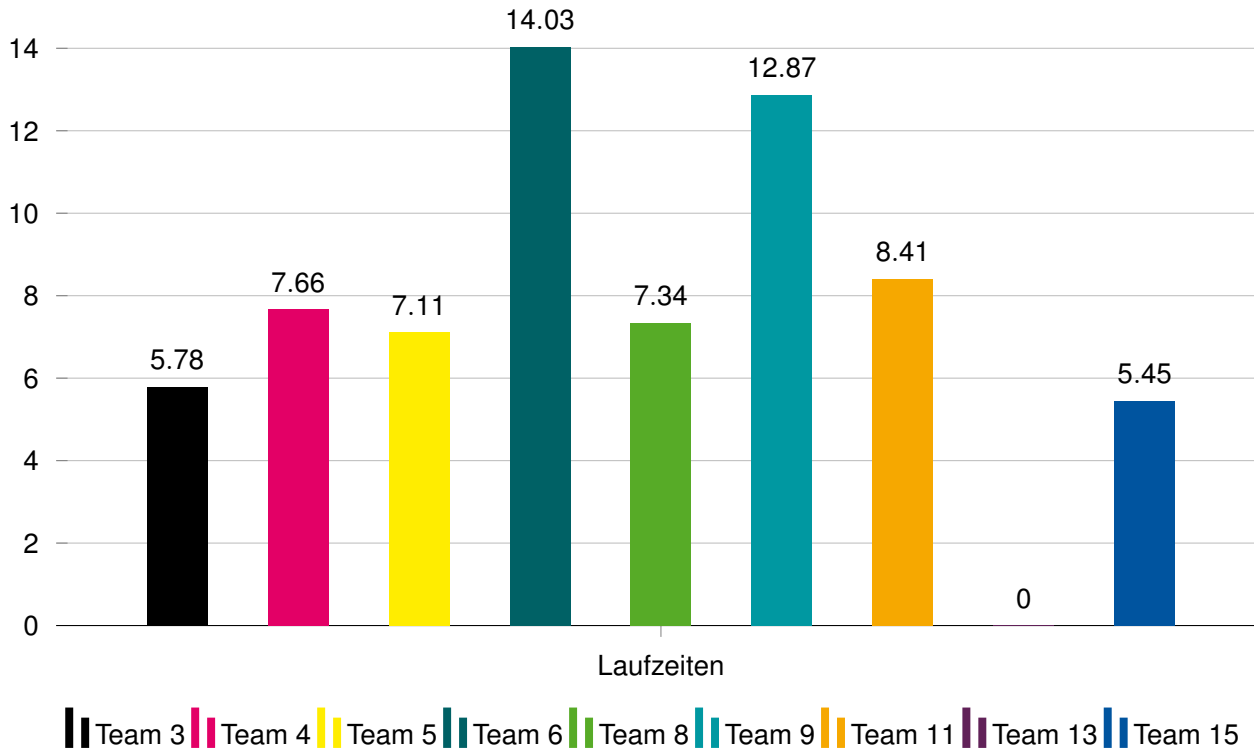
Siegerehrung



Platz 3:
Team 5

Platz 2:
Team 3

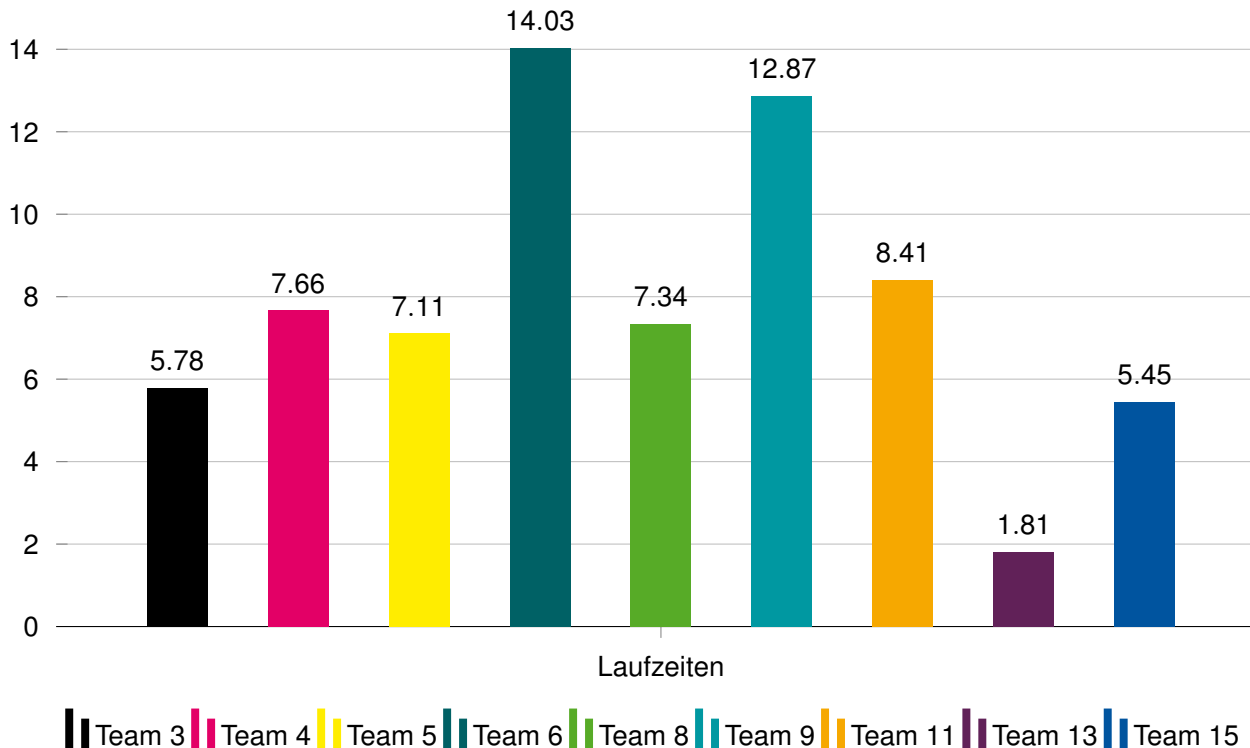
Siegerehrung



Platz 3:
Team 5

Platz 2:
Team 3

Siegerehrung



Platz 3:
Team 5

Platz 2:
Team 3

And the winner is ...
Team 13!

Sieger

Platz 1:

René Schäfer, Johannes Seiffarth, Leon Wittwer mit “WeirdTrigramStorage”

Platz 2:

Eric Wagner mit “ComplexTrigramStorage”

Platz 3:

Team 5 mit “TrieTrigramStorage”