Introduction

- A field’s values are stored in one of several data structures
  - The data structure depends on the field’s data type
- Ensures efficient data access — e.g. searches
- Handled by Apache Lucene, not Elasticsearch
- This lecture focuses on inverted indices

CodingExplained
CODE. SLEEP. REPEAT.

Inverted indices

- Mapping between terms and which documents contain them
- Outside the context of analyzers, we use the terminology “terms”
- Terms are sorted alphabetically
- Inverted indices contain more than just terms and document IDs
  - E.g. information for relevance scoring
- One inverted index per text field
- Other data types use BKD trees, for instance

Lecture summary (1/2)

- Values for a text field are analyzed and the results are stored within an inverted index
- Each field has a dedicated inverted index
- An inverted index is a mapping between terms and which documents contain them
- Terms are sorted alphabetically for performance reasons
- Created and maintained by Apache Lucene, *not* Elasticsearch

Lecture summary (2/2)

- Inverted indices enable **fast** searches
- Inverted indices contain other data as well
  - E.g. things used for relevance scoring (covered later)
- Elasticsearch (technically, Apache Lucene) uses other data structures as well
  - E.g. BKD trees for numeric values, dates, and geospatial data