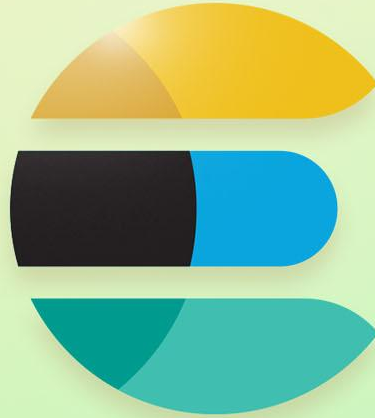# Disjunction max (`dis_max`)

The best matching field's relevance score is used

dis_max query

Query

```
GET /products/_search
{
  "query": {
    "dis_max": {
      "queries": [
        { "match": { "name": "vegetable" } },
        { "match": { "tags": "vegetable" } }
      ]
    }
  }
}
```

Document

```
{
    "name": "Vegetable Broth", (2.7086513)
    "tags": ["Vegetable"] (5.6236019)
}
```

Search results

```
{
    "_score": 5.6236019,
    "_source": {
        "name": "Vegetable Broth",
        "tags": [
            "Vegetable"
        ]
    }
}
```

CodingExplained
CODE. SLEEP. REPEAT.

Only one query clause matches

Query

```
GET /products/_search
{
  "query": {
    "dis_max": {
      "queries": [
        { "match": { "name": "vegetable" } },
        { "match": { "tags": "vegetable" } }
      ]
    }
  }
}
```

Document

```
{
    "name": "Radish", N/A
    "tags": ["Vegetable"] (5.6236019)
}
```

Search results

```
{
    "_score": 5.6236019,
    "_source": {
        "name" : "Radish",
        "tags" : [
            "Vegetable"
        ]
    }
}
```

CodingExplained
CODE. SLEEP. REPEAT.

tie_breaker parameter

Query

```
GET /products/_search
{
  "query": {
    "dis_max": {
      "queries": [
        { "match": { "name": "vegetable" } },
        { "match": { "tags": "vegetable" } }
      ],
      "tie_breaker": 0.3
    }
  }
}
```

Document

```
{
  "name": "Vegetable Broth", (0.8125954) 🔵
  "tags": ["Vegetable"] (5.6236019)
}
```

🔵 **Score calculation**

2.7086513 x 0.3 = 0.8125954

Search results

```
{
  "_score": 6.4361973, 🔴
  "_source": {
    "name": "Vegetable Broth",
    "tags": [
      "Vegetable"
    ]
  }
}
```

🔴 **Score calculation**

5.6236019 + 0.8125954 = 6.4361973

CodingExplained
CODE. SLEEP. REPEAT.

How `multi_match` works*
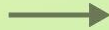
```
GET /products/_search
{
  "query": {
    "multi_match": {
      "query": "vegetable",
      "fields": ["name", "tags"]
    }
  }
}
```

→

```
GET /products/_search
{
  "query": {
    "dis_max": {
      "queries": [
        { "match": { "name": "vegetable" } },
        { "match": { "tags": "vegetable" } }
      ]
    }
  }
}
```

CodingExplained
CODE. SLEEP. REPEAT.

How `multi_match` works

```
GET /products/_search
{
  "query": {
    "multi_match": {
      "query": "vegetable",
      "fields": ["name", "tags"],
      "tie_breaker": 0.3
    }
  }
}
```

→

```
GET /products/_search
{
  "query": {
    "dis_max": {
      "queries": [
        { "match": { "name": "vegetable" } },
        { "match": { "tags": "vegetable" } }
      ],
      "tie_breaker": 0.3
    }
  }
}
```

CodingExplained
CODE. SLEEP. REPEAT.

Lecture summary

- The `dis_max` query is a compound query

  - A document matches if at least one leaf query matches

- The best matching matching query clause's relevance score is used for a

  document's `_score`

- `tie_breaker` can be used to "reward" documents that match multiple queries

- `multi_match` queries are often translated into `dis_max` queries internally