|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MinimumSnippet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | |  |
| PREV CLASS   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?MinimumSnippet.html)    [**NO FRAMES**](http://docs.google.com/MinimumSnippet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## Class MinimumSnippet

java.lang.Object  
 **MinimumSnippet**

public class **MinimumSnippet**extends java.lang.Object

When you do a web search, the results page shows you a [snippet](http://searchengineland.com/anatomy-of-a-google-snippet-38357) for each result, showing you search terms in context. For purposes of this project, a snippet is a subsequence of a document that contains all the search terms. For this project, you will write code that, given a document (a sequence of words) and set of search terms, find the minimal length subsequence in the document that contains all of the search terms. If there are multiple subsequences that have the same minimal length, you may return any one of them.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**MinimumSnippet**](http://docs.google.com/MinimumSnippet.html#MinimumSnippet(java.lang.Iterable,%20java.util.List))(java.lang.Iterable<java.lang.String> document, java.util.List<java.lang.String> terms)            Compute minimum snippet. |

|  |  |
| --- | --- |
| **Method Summary** | |
| boolean | [**foundAllTerms**](http://docs.google.com/MinimumSnippet.html#foundAllTerms())()            Returns whether or not all terms were found in the document. |
| int | [**getEndingPos**](http://docs.google.com/MinimumSnippet.html#getEndingPos())()            Return the ending position of the snippet |
| int | [**getLength**](http://docs.google.com/MinimumSnippet.html#getLength())()            Return total number of elements contained in the snippet. |
| int | [**getPos**](http://docs.google.com/MinimumSnippet.html#getPos(int))(int index)            Returns the position of one of the search terms as it appears in the original document |
| int | [**getStartingPos**](http://docs.google.com/MinimumSnippet.html#getStartingPos())()            Return the starting position of the snippet |

|  |
| --- |
| **Methods inherited from class java.lang.Object** |
| equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

|  |
| --- |
| **Constructor Detail** |

### MinimumSnippet

public **MinimumSnippet**(java.lang.Iterable<java.lang.String> document,  
 java.util.List<java.lang.String> terms)

Compute minimum snippet. Given a document (represented as an List), and a set of terms (represented as a List), find the shortest subsequence of the document that contains all of the terms. This constructor should find the minimum snippet, and store information about the snippet in fields so that the methods can be called to query information about the snippet. All significant computation should be done during construction.

**Parameters:**document - The Document is an Iterable. Do not change the document. It is an Iterable, rather than a List, to allow for implementations where we scan very large documents that are not read entirely into memory. If you have problems figuring out how to solve it with the document represented as an Iterable, you may cast it to a List; in all but a very small number of test cases, in will in fact be a List.terms - The terms you need to look for. The terms will be unique (e.g., no term will be repeated), although you do not need to check for that. There should always be at least one term and your code should throw an IllegalArgumentException if "terms" is empty.

|  |
| --- |
| **Method Detail** |

### foundAllTerms

public boolean **foundAllTerms**()

Returns whether or not all terms were found in the document. If all terms were not found, then none of the other methods should be called.

**Returns:**whether all terms were found in the document.

### getStartingPos

public int **getStartingPos**()

Return the starting position of the snippet

**Returns:**the index in the document of the first element of the snippet

### getEndingPos

public int **getEndingPos**()

Return the ending position of the snippet

**Returns:**the index in the document of the last element of the snippet

### getLength

public int **getLength**()

Return total number of elements contained in the snippet.

**Returns:**

### getPos

public int **getPos**(int index)

Returns the position of one of the search terms as it appears in the original document

**Parameters:**index - index of the term in the original list of terms. For example, if index is 0 then the method will return the position (in the document) of the first search term. If the index is 1, then the method will return the position (in the document) of the second search term. Etc.

EXAMPLE:

Document = A C B A C

Terms = A B

getPos(0) is 3 because term 0 is the A, and the A found in the shortest snippet is located at position 3 in the original document.

**Returns:**position of the term in the document

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MinimumSnippet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | |  |
| PREV CLASS   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?MinimumSnippet.html)    [**NO FRAMES**](http://docs.google.com/MinimumSnippet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[Web Accessibility](https://www.umd.edu/web-accessibility)