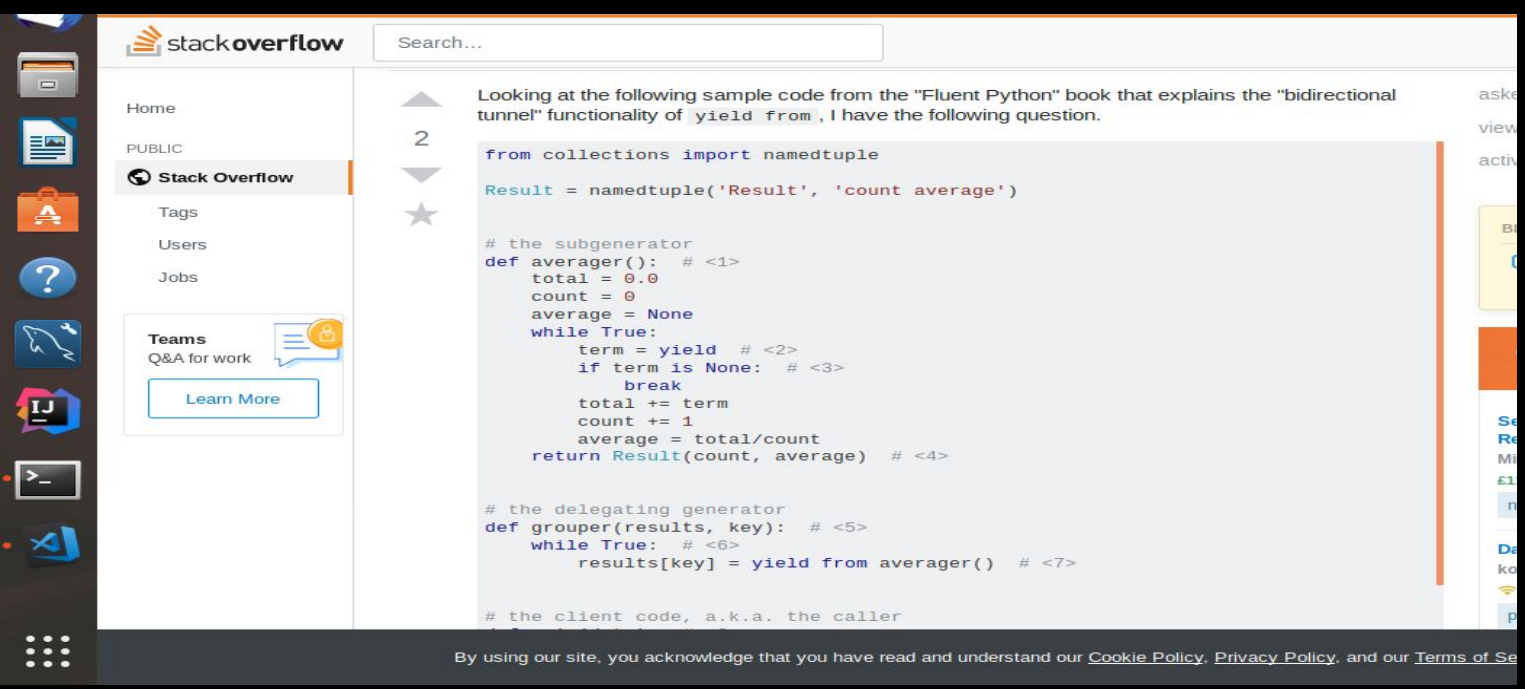
Ajit Jadhav S20160010034
Arvind Deshraj S20160010007
Junaid N Z S20160010036
Rohan S S20160010073

Code Search

By Stallions

What are the uses of code search?



Makes developer's life easier.
Find better implementations of existing codes.
Able to find snippets of code inside large codebase

IR for the rescue

Lucene!

- Wildcard queries
- K-gram index
- Natural language queries

Procedure

1. Every query can be processed through different pipe lines. Either subjected to standard tokenization and send for search or ..

2. Queries may also be placed as wildcard queries.

3. The documents are indexed as a normal inverted posting list and also as an n-gram positional inverted posting list.

Indexing

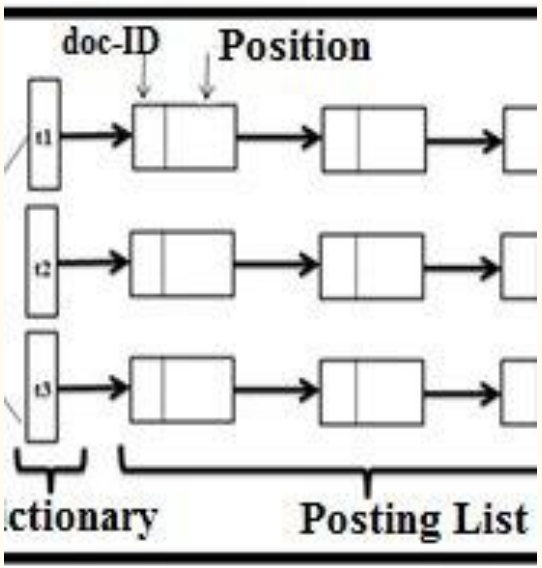
1. We have 260 C-codes as our entire collection.(~400KB)
2. These documents are passed through our index builder codes(n-gram and normal).
3. These documents are then indexed depending on which function we chose as either a positional inverted index(~300KB) or as an n-gram positional inverted index.
4. We make use of the standard analyser to find stop words and index the documents.

Positional Inverted Index

to, 993427:
⟨ 1, 6: ⟨7, 18, 33, 72, 86, 231⟩;
2, 5: ⟨1, 17, 74, 222, 255⟩;
4, 5: ⟨8, 16, 190, 429, 433⟩;
5, 2: ⟨363, 367⟩;
7, 3: ⟨13, 23, 191⟩; . . . ⟩

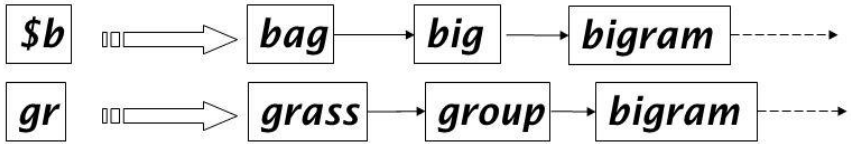
be, 178239:
⟨ 1, 2: ⟨17, 25⟩;
4, 5: ⟨17, 191, 291, 430, 434⟩;
5, 3: ⟨14, 19, 101⟩; . . . ⟩

► **Figure 2.1** Positional index example. The word to has a document frequency 993,477, and occurs 6 times in document 1 at positions 7, 18, 33, etc.



K-gram Index

Bigram index example



A k-gram index is an index in which the dictionary consists of all k-grams that occur in any word in the lexicon

Each postings list point from the k-gram to all lexicon words containing that k-gram.

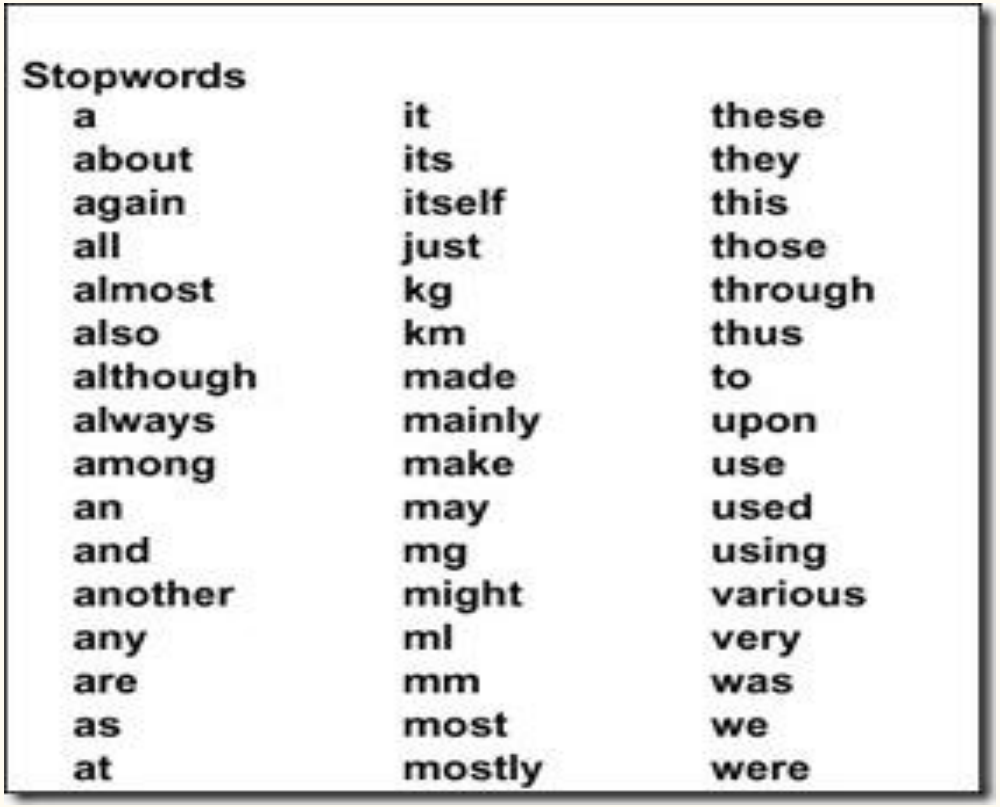
Hypothesis

K-grams will retrieve more documents as it will have lot of code snippets that match.

- **Eg - if we search for “arrange all elements in an array” we will get results ranging from normal sorting algorithms to complex graph implementations which have sorting in them.**

Query processing

Stop-word removal



Stopwords		
a	it	these
about	its	they
again	itself	this
all	just	those
almost	kg	through
also	km	thus
although	made	to
always	mainly	upon
among	make	use
an	may	used
and	mg	using
another	might	various
any	ml	very
are	mm	was
as	most	we
at	mostly	were

- These are generally words that don’t give value towards the search. This is because these set of words(only a small example is shown here) appear in almost all documents.

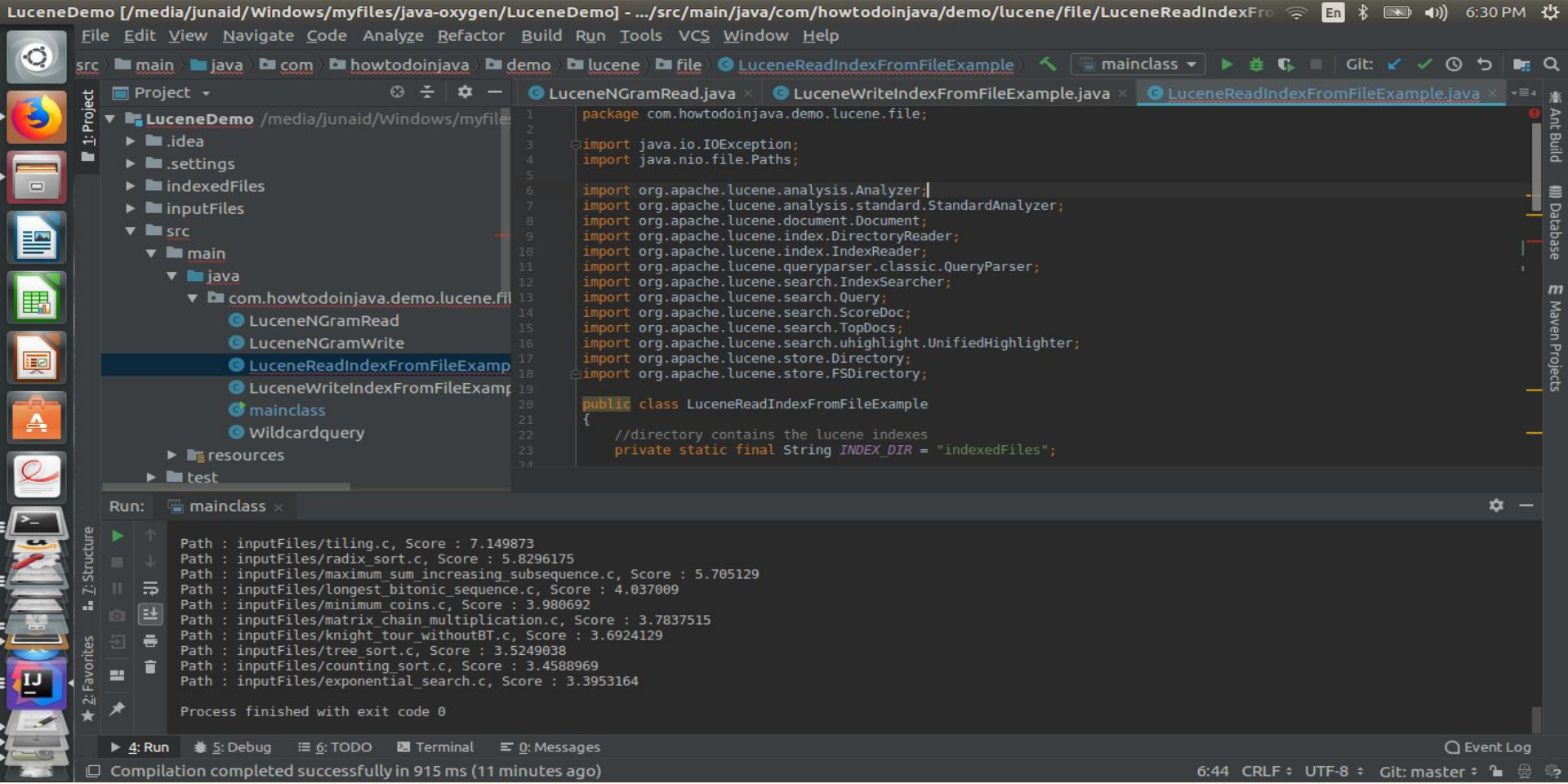
Stemming

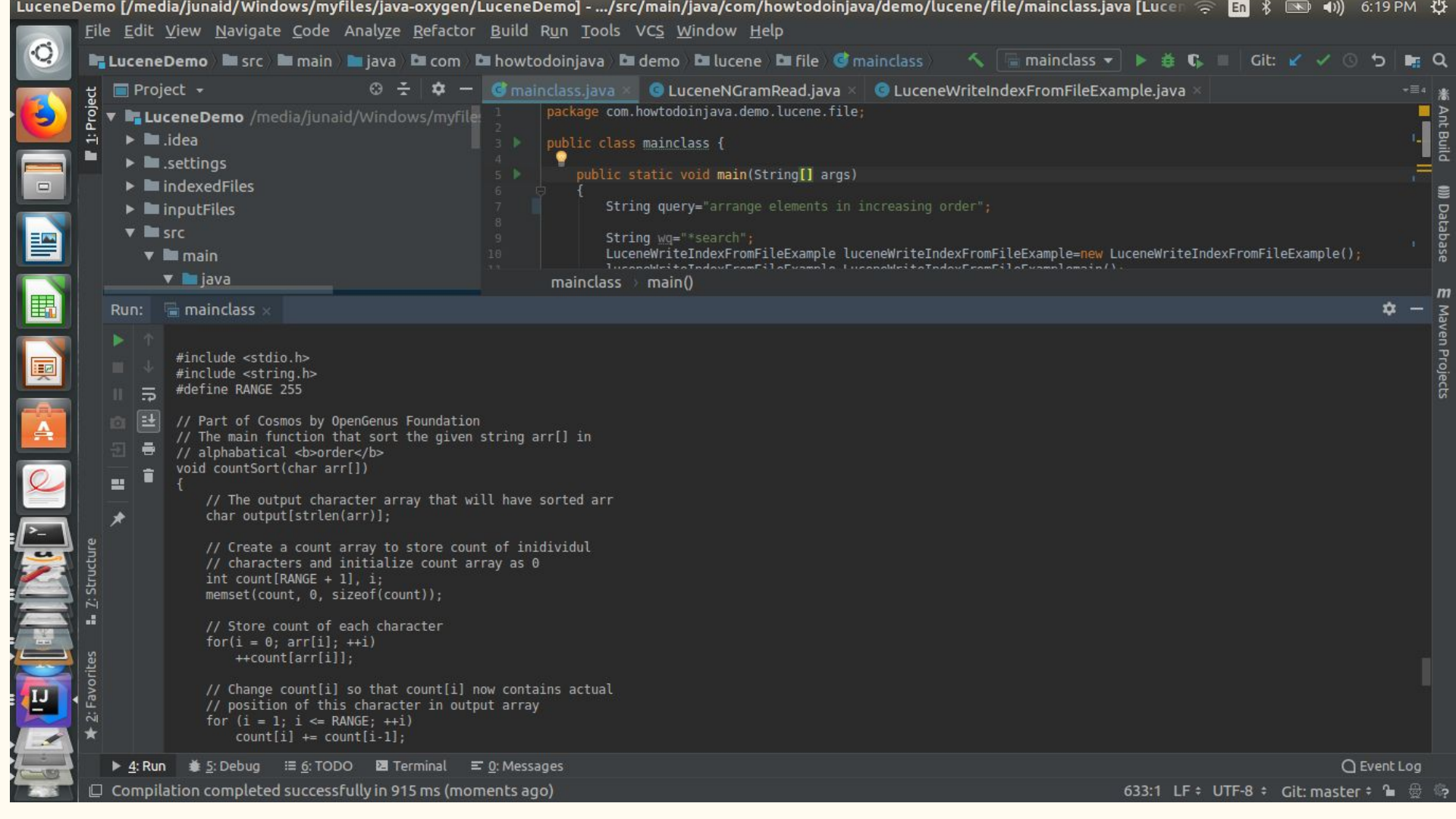
```
>>> print(stemmer.stem("running"))
run
```

- Stemming follows different set of rules based on what algorithm we follow.
- For the purpose of the project we use the famous snowball stemmer

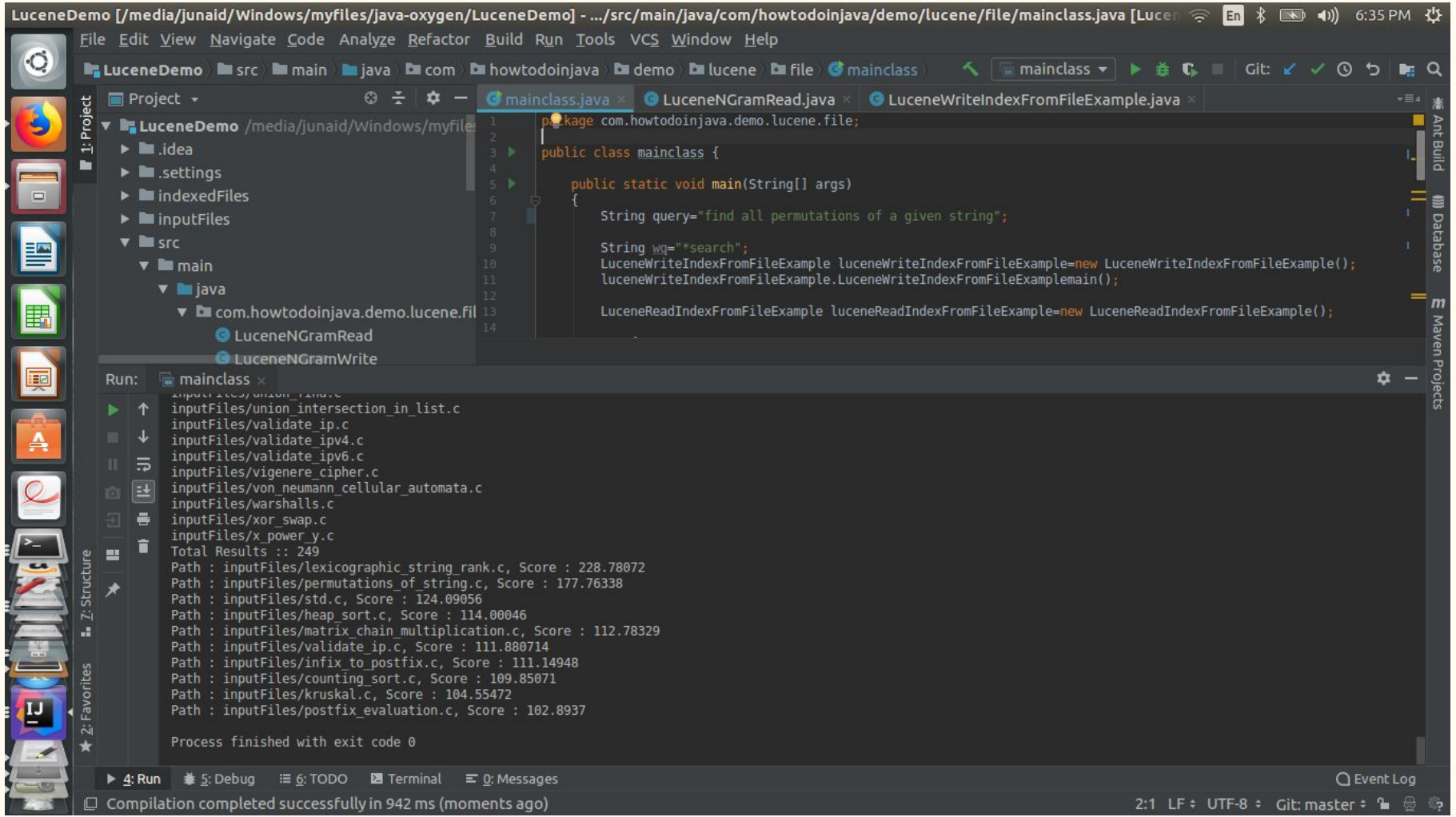
Few results of the CodeSearch

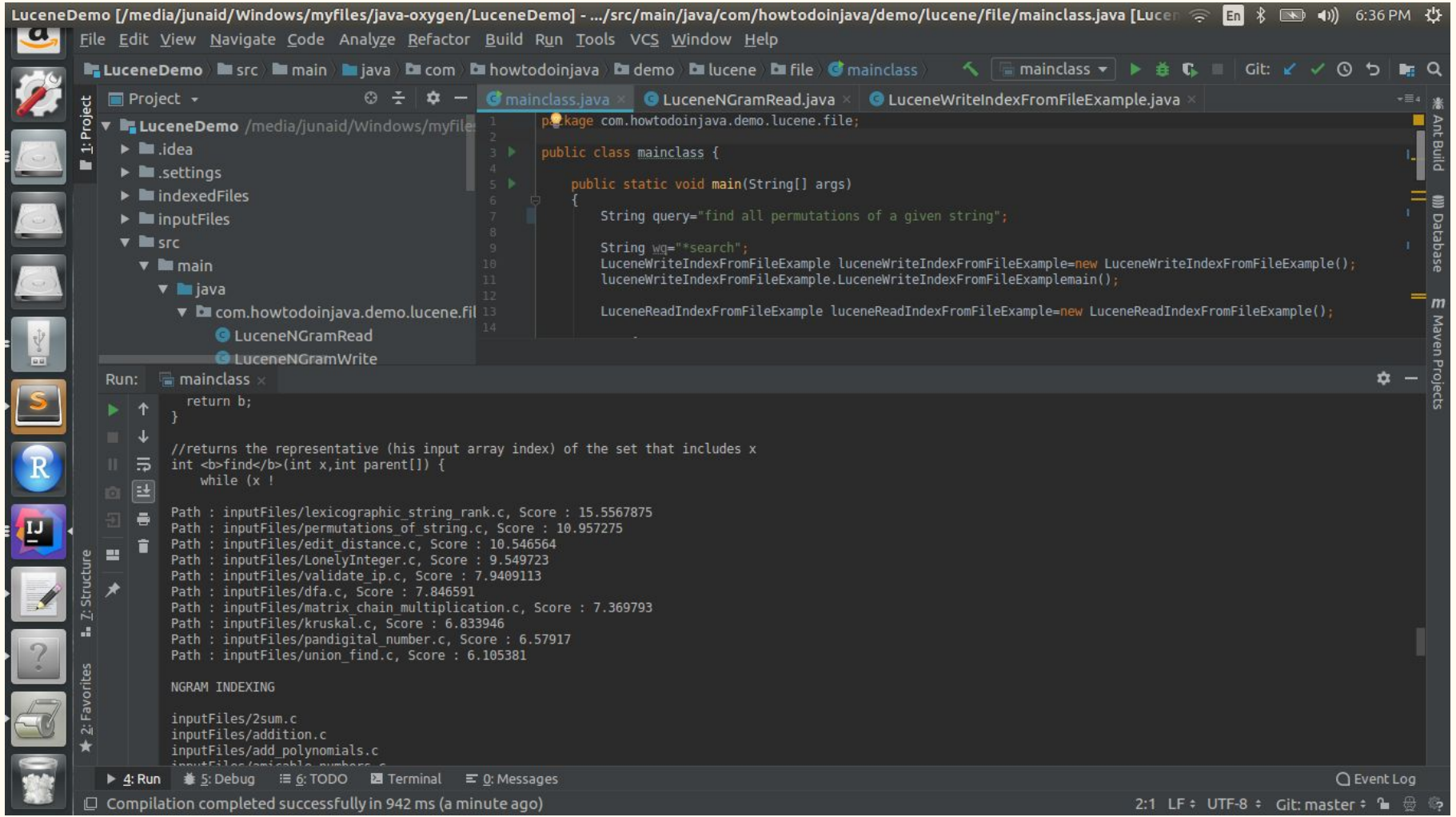
Query : Arrange elements in increasing order


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/LuceneReadIndexFro
6:30 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
src main java com howtodoinjava demo lucene file LuceneReadIndexFromFileExample
mainclass
Git:
Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
LuceneReadIndexFromFileExamp
LuceneWriteIndexFromFileExamp
mainclass
Wildcardquery
resources
test
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
LuceneReadIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 import java.io.IOException;
4 import java.nio.file.Paths;
5
6 import org.apache.lucene.analysis.Analyzer;
7 import org.apache.lucene.analysis.standard.StandardAnalyzer;
8 import org.apache.lucene.document.Document;
9 import org.apache.lucene.index.DirectoryReader;
10 import org.apache.lucene.index.IndexReader;
11 import org.apache.lucene.queryparser.classic.QueryParser;
12 import org.apache.lucene.search.IndexSearcher;
13 import org.apache.lucene.search.Query;
14 import org.apache.lucene.search.ScoreDoc;
15 import org.apache.lucene.search.TopDocs;
16 import org.apache.lucene.search.uhighlight.UnifiedHighlighter;
17 import org.apache.lucene.store.Directory;
18 import org.apache.lucene.store.FSDirectory;
19
20 public class LuceneReadIndexFromFileExample
21 {
22 //directory contains the lucene indexes
23 private static final String INDEX_DIR = "indexedFiles";
Run: mainclass
Path : inputFiles/tiling.c, Score : 7.149873
Path : inputFiles/radix_sort.c, Score : 5.8296175
Path : inputFiles/maximum_sum_increasing_subsequence.c, Score : 5.705129
Path : inputFiles/longest_bitonic_sequence.c, Score : 4.037009
Path : inputFiles/minimum_coins.c, Score : 3.980692
Path : inputFiles/matrix_chain_multiplication.c, Score : 3.7837515
Path : inputFiles/knight_tour_withoutBT.c, Score : 3.6924129
Path : inputFiles/tree_sort.c, Score : 3.5249038
Path : inputFiles/counting_sort.c, Score : 3.4588969
Path : inputFiles/exponential_search.c, Score : 3.3953164
Process finished with exit code 0
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 915 ms (11 minutes ago)
6:44 CRLF UTF-8 Git: master
Ant Build
Database
Maven Projects
1: Project
7: Structure
2: Favorites

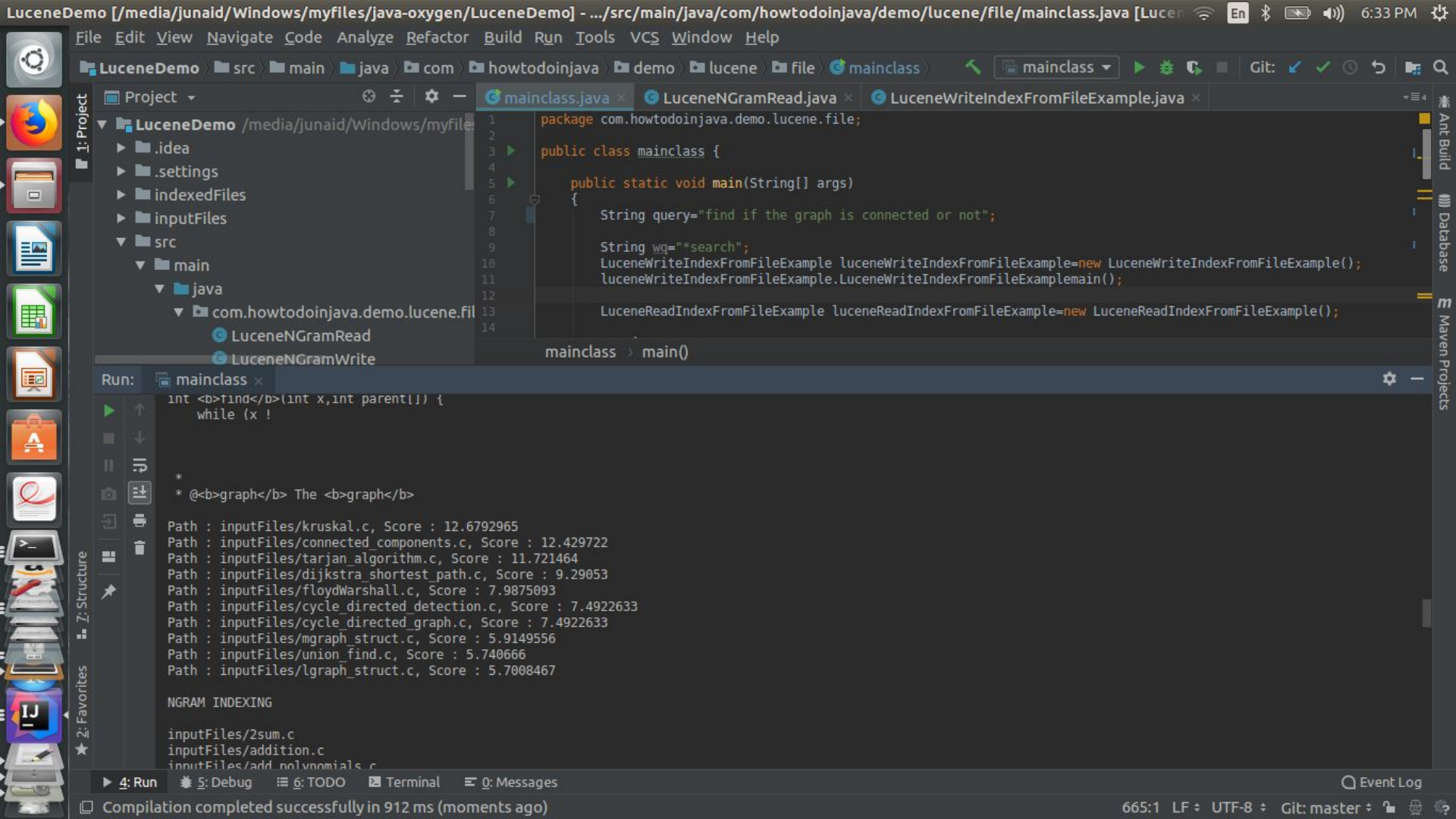

LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucer
6:19 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
1: Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
package com.howtodoinjava.demo.lucene.file;
public class mainclass {
public static void main(String[] args)
{
String query="arrange elements in increasing order";
String wq="*search";
LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
mainclass › main()
Run:
mainclass
#include <stdio.h>
#include <string.h>
#define RANGE 255
// Part of Cosmos by OpenGenus Foundation
// The main function that sort the given string arr[] in
// alphabatical order
void countSort(char arr[])
{
// The output character array that will have sorted arr
char output[strlen(arr)];
// Create a count array to store count of inidividul
// characters and initialize count array as 0
int count[RANGE + 1], i;
memset(count, 0, sizeof(count));
// Store count of each character
for(i = 0; arr[i]; ++i)
++count[arr[i]];
// Change count[i] so that count[i] now contains actual
// position of this character in output array
for (i = 1; i <= RANGE; ++i)
count[i] += count[i-1];
7: Structure
2: Favorites
Ant Build
Database
Maven Projects
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 915 ms (moments ago)
633:1 LF UTF-8 Git: master

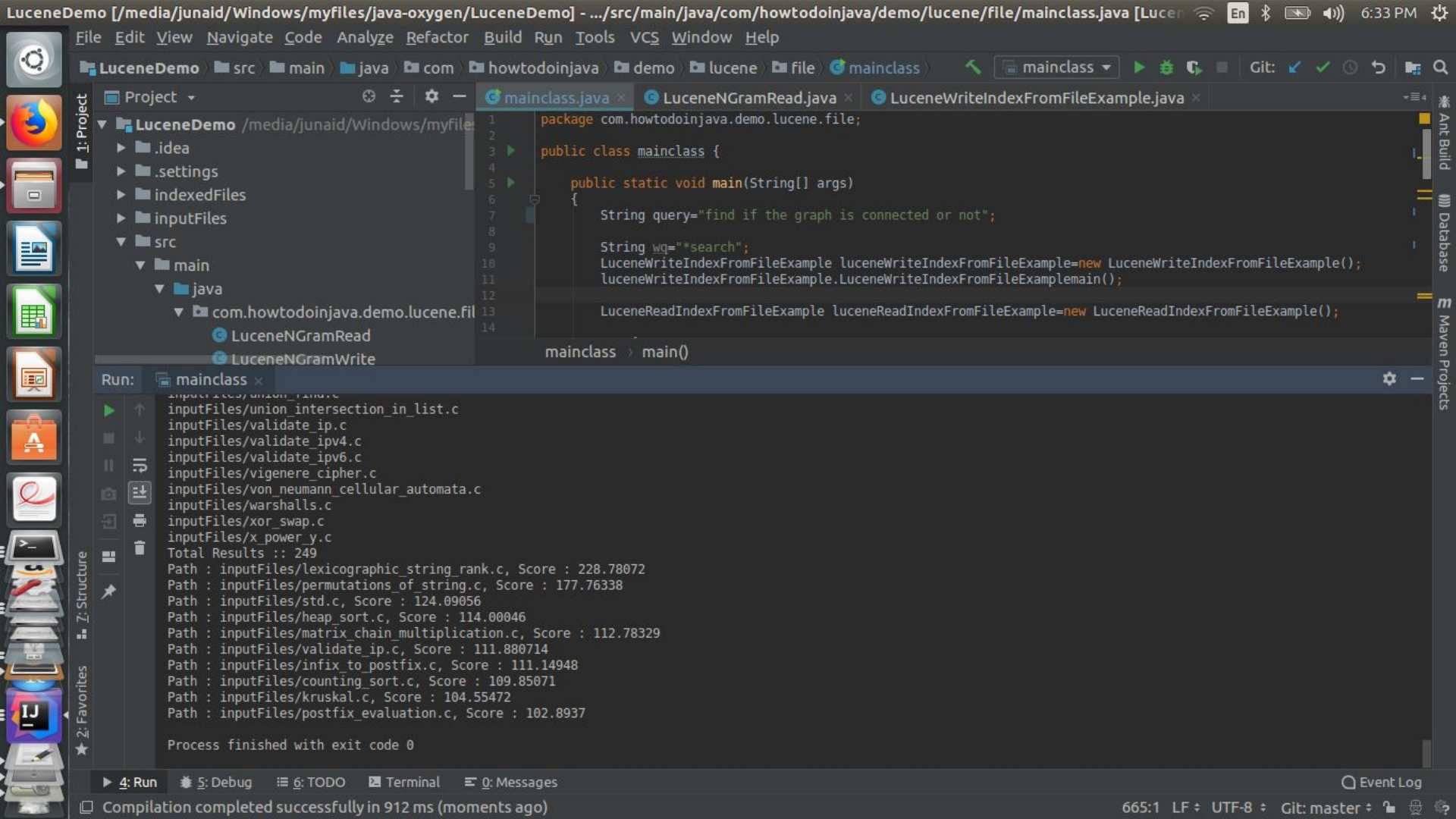
Query : Code for string permutations.

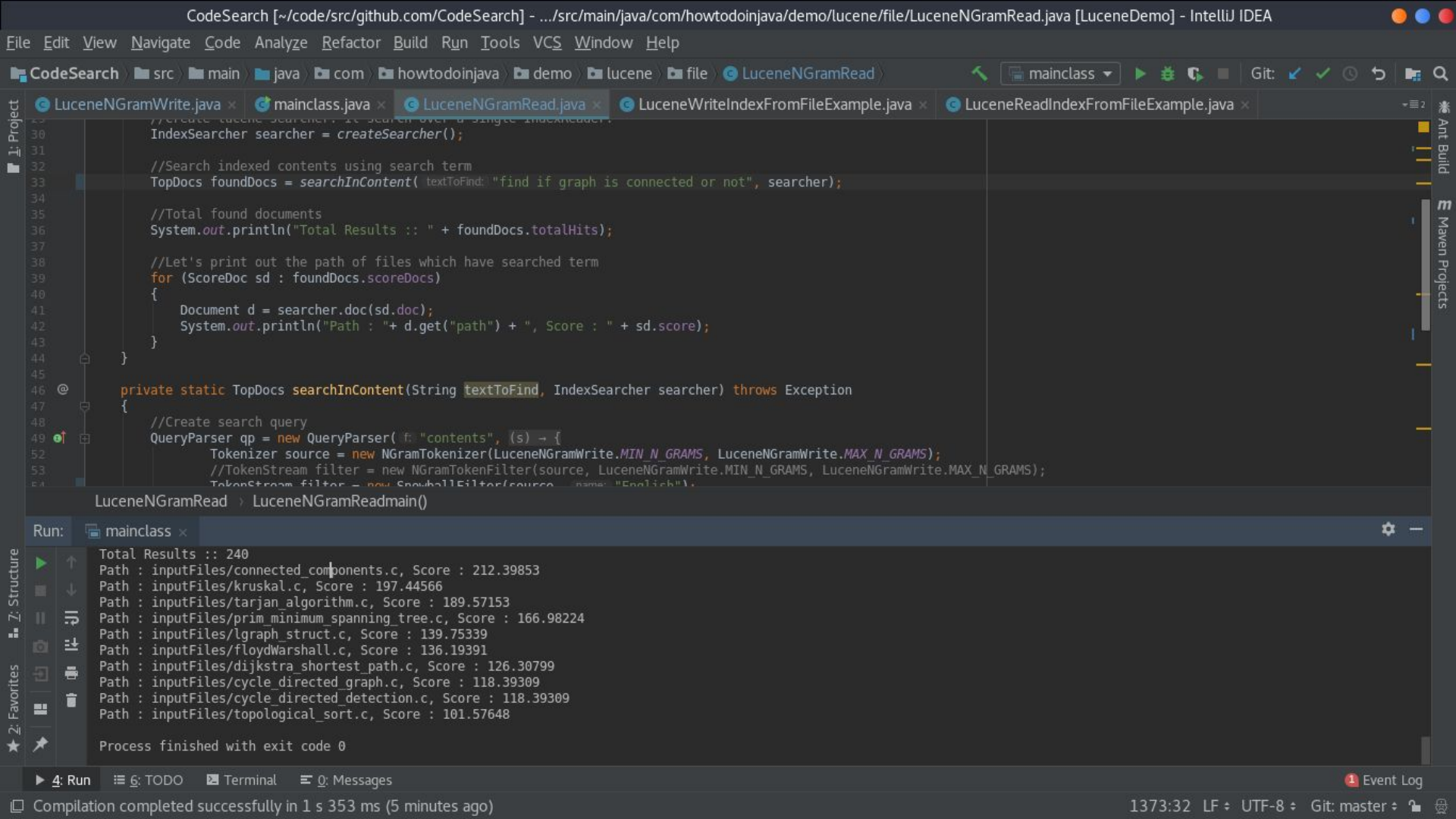

LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:35 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find all permutations of a given string";
8
9 String wq="*search";
10 LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
Run: mainclass
inputFiles/union_intersection_in_list.c
inputFiles/validate_ip.c
inputFiles/validate_ipv4.c
inputFiles/validate_ipv6.c
inputFiles/vigenere_cipher.c
inputFiles/von_neumann_cellular_automata.c
inputFiles/warshalls.c
inputFiles/xor_swap.c
inputFiles/x_power_y.c
Total Results :: 249
Path : inputFiles/lexicographic_string_rank.c, Score : 228.78072
Path : inputFiles/permutations_of_string.c, Score : 177.76338
Path : inputFiles/std.c, Score : 124.09056
Path : inputFiles/heap_sort.c, Score : 114.00046
Path : inputFiles/matrix_chain_multiplication.c, Score : 112.78329
Path : inputFiles/validate_ip.c, Score : 111.880714
Path : inputFiles/infix_to_postfix.c, Score : 111.14948
Path : inputFiles/counting_sort.c, Score : 109.85071
Path : inputFiles/kruskal.c, Score : 104.55472
Path : inputFiles/postfix_evaluation.c, Score : 102.8937
Process finished with exit code 0
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 942 ms (moments ago)
2:1 LF UTF-8 Git: master
1: Project
7: Structure
2: Favorites
Ant Build
Database
Maven Projects


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:36 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
package com.howtodoinjava.demo.lucene.file;
public class mainclass {
public static void main(String[] args)
{
String query="find all permutations of a given string";
String wq="*search";
LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
Run: mainclass
return b;
}
//returns the representative (his input array index) of the set that includes x
int find(int x,int parent[]) {
while (x !
Path : inputFiles/lexicographic_string_rank.c, Score : 15.5567875
Path : inputFiles/permutations_of_string.c, Score : 10.957275
Path : inputFiles/edit_distance.c, Score : 10.546564
Path : inputFiles/LonelyInteger.c, Score : 9.549723
Path : inputFiles/validate_ip.c, Score : 7.9409113
Path : inputFiles/dfa.c, Score : 7.846591
Path : inputFiles/matrix_chain_multiplication.c, Score : 7.369793
Path : inputFiles/kruskal.c, Score : 6.833946
Path : inputFiles/pandigital_number.c, Score : 6.57917
Path : inputFiles/union_find.c, Score : 6.105381
NGRAM INDEXING
inputFiles/2sum.c
inputFiles/addition.c
inputFiles/add_polynomials.c
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 942 ms (a minute ago)
2:1 LF UTF-8 Git: master
1: Project
2: Favorites
7: Structure
Ant Build
Database
Maven Projects

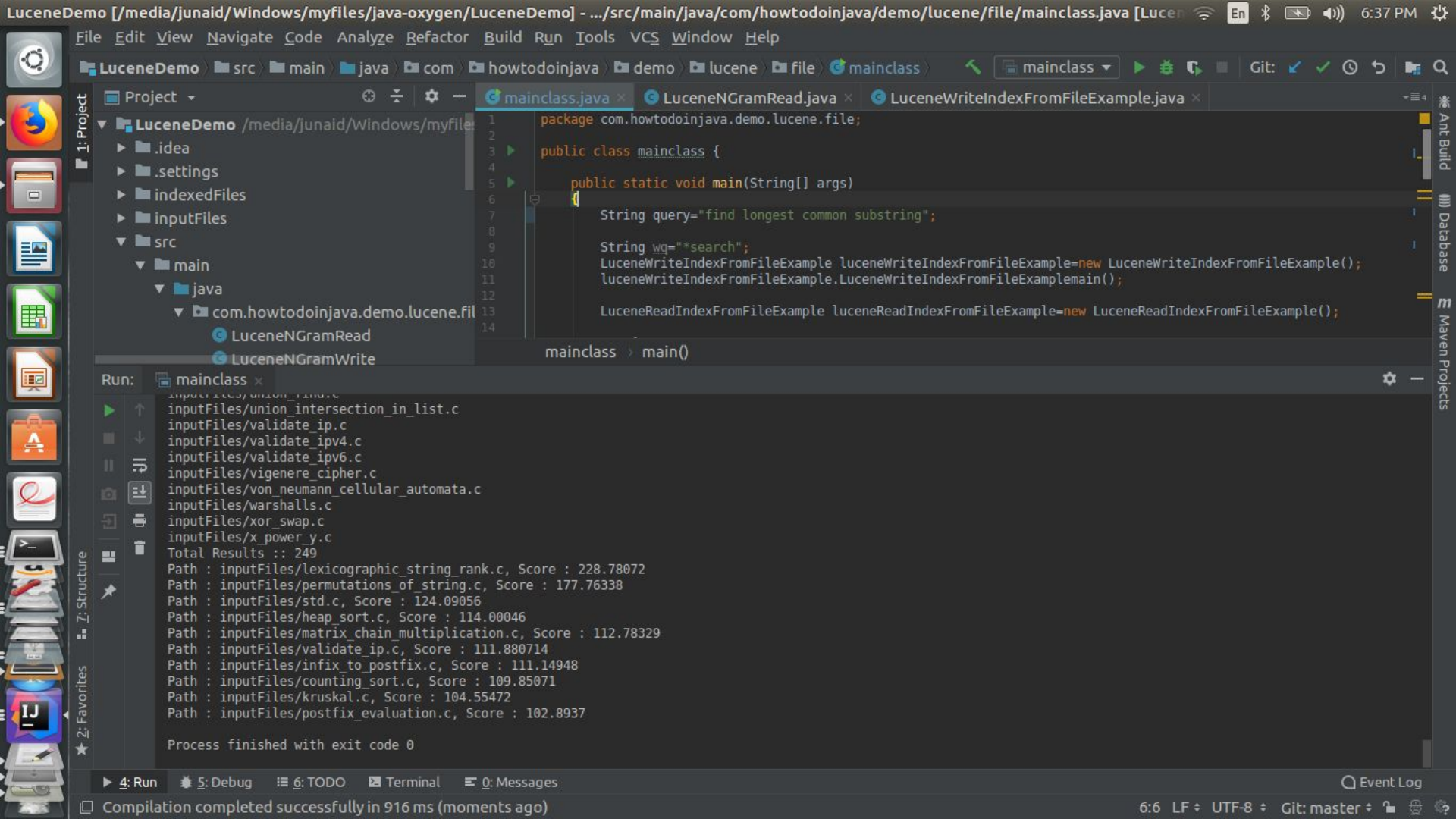
Query : Check if a graph is connected or not


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:33 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find if the graph is connected or not";
8
9 String wq="*search";
10 LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
mainclass › main()
Run: mainclass
int find(int x,int parent[]) {
while (x !
*
* @graph The graph
Path : inputFiles/kruskal.c, Score : 12.6792965
Path : inputFiles/connected_components.c, Score : 12.429722
Path : inputFiles/tarjan_algorithm.c, Score : 11.721464
Path : inputFiles/dijkstra_shortest_path.c, Score : 9.29053
Path : inputFiles/floydWarshall.c, Score : 7.9875093
Path : inputFiles/cycle_directed_detection.c, Score : 7.4922633
Path : inputFiles/cycle_directed_graph.c, Score : 7.4922633
Path : inputFiles/mgraph_struct.c, Score : 5.9149556
Path : inputFiles/union_find.c, Score : 5.740666
Path : inputFiles/lgraph_struct.c, Score : 5.7008467
NGRAM INDEXING
inputFiles/2sum.c
inputFiles/addition.c
inputFiles/add_polynomials.c
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 912 ms (moments ago)
665:1 LF UTF-8 Git: master
Ant Build
Database
Maven Projects
1: Project
7: Structure
2: Favorites

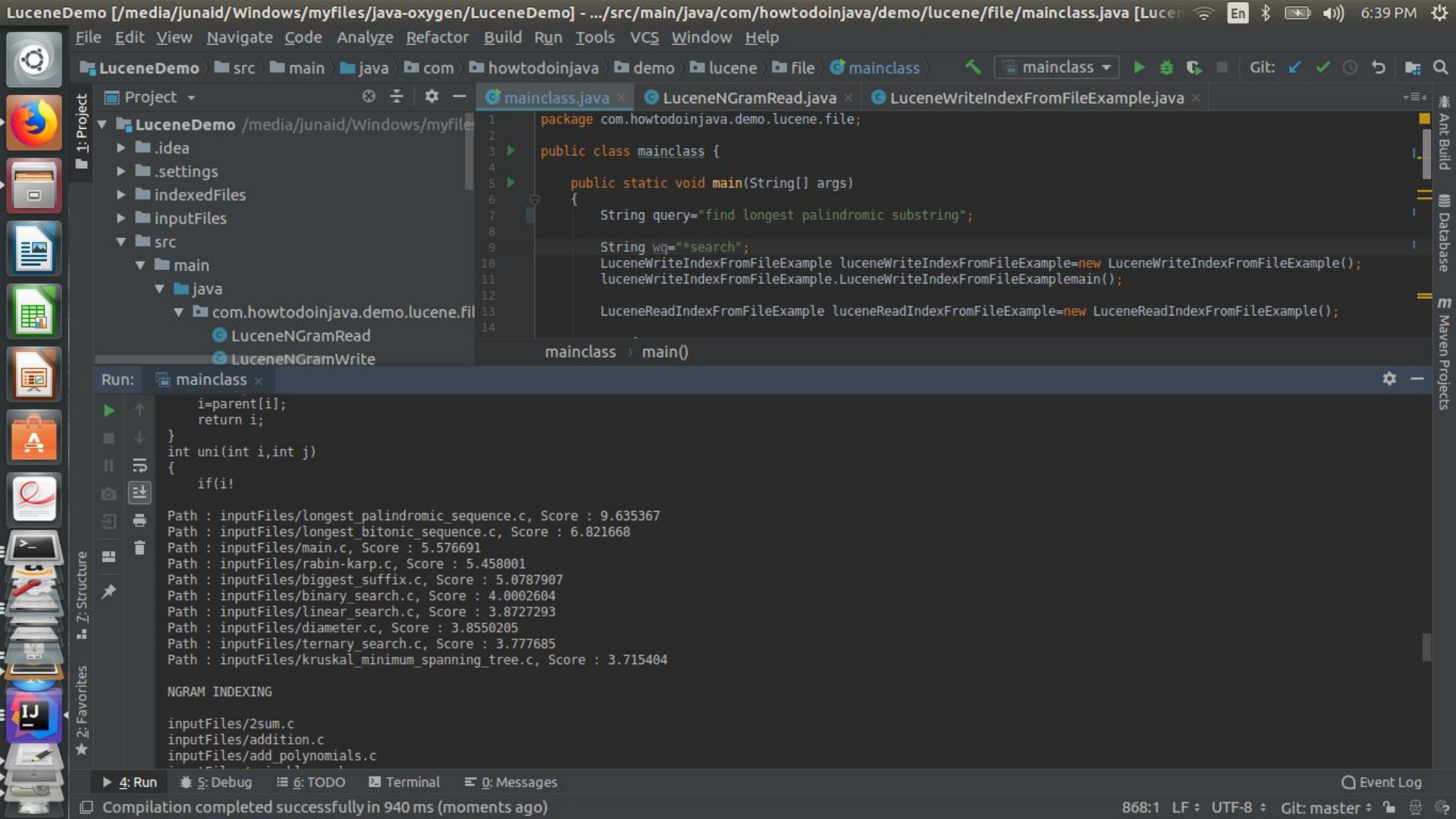

LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:33 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find if the graph is connected or not";
8
9 String wq="*search";
10 LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
mainclass › main()
Run: mainclass
inputFiles/union_intersection_in_list.c
inputFiles/validate_ip.c
inputFiles/validate_ipv4.c
inputFiles/validate_ipv6.c
inputFiles/vigenere_cipher.c
inputFiles/von_neumann_cellular_automata.c
inputFiles/warshalls.c
inputFiles/xor_swap.c
inputFiles/x_power_y.c
Total Results :: 249
Path : inputFiles/lexicographic_string_rank.c, Score : 228.78072
Path : inputFiles/permutations_of_string.c, Score : 177.76338
Path : inputFiles/std.c, Score : 124.09056
Path : inputFiles/heap_sort.c, Score : 114.00046
Path : inputFiles/matrix_chain_multiplication.c, Score : 112.78329
Path : inputFiles/validate_ip.c, Score : 111.880714
Path : inputFiles/infix_to_postfix.c, Score : 111.14948
Path : inputFiles/counting_sort.c, Score : 109.85071
Path : inputFiles/kruskal.c, Score : 104.55472
Path : inputFiles/postfix_evaluation.c, Score : 102.8937
Process finished with exit code 0
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 912 ms (moments ago)
665:1 LF UTF-8 Git: master
1: Project
7: Structure
2: Favorites
Ant Build
Database
Maven Projects


CodeSearch [~/code/src/github.com/CodeSearch] - .../src/main/java/com/howtodoinjava/demo/lucene/file/LuceneNGramRead.java [LuceneDemo] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CodeSearch src main java com howtodoinjava demo lucene file LuceneNGramRead
mainclass
Git:
LuceneNGramWrite.java
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
LuceneReadIndexFromFileExample.java
IndexSearcher searcher = createSearcher();
//Search indexed contents using search term
TopDocs foundDocs = searchInContent(textToFind: "find if graph is connected or not", searcher);
//Total found documents
System.out.println("Total Results :: " + foundDocs.totalHits);
//Let's print out the path of files which have searched term
for (ScoreDoc sd : foundDocs.scoreDocs)
{
Document d = searcher.doc(sd.doc);
System.out.println("Path : "+ d.get("path") + ", Score : " + sd.score);
}
}
private static TopDocs searchInContent(String textToFind, IndexSearcher searcher) throws Exception
{
//Create search query
QueryParser qp = new QueryParser(f: "contents", (s) → {
Tokenizer source = new NGramTokenizer(LuceneNGramWrite.MIN_N_GRAMS, LuceneNGramWrite.MAX_N_GRAMS);
//TokenStream filter = new NGramTokenFilter(source, LuceneNGramWrite.MIN_N_GRAMS, LuceneNGramWrite.MAX_N_GRAMS);
LuceneNGramRead › LuceneNGramReadmain()
Run: mainclass
Total Results :: 240
Path : inputFiles/connected_components.c, Score : 212.39853
Path : inputFiles/kruskal.c, Score : 197.44566
Path : inputFiles/tarjan_algorithm.c, Score : 189.57153
Path : inputFiles/prim_minimum_spanning_tree.c, Score : 166.98224
Path : inputFiles/lgraph_struct.c, Score : 139.75339
Path : inputFiles/floydWarshall.c, Score : 136.19391
Path : inputFiles/dijkstra_shortest_path.c, Score : 126.30799
Path : inputFiles/cycle_directed_graph.c, Score : 118.39309
Path : inputFiles/cycle_directed_detection.c, Score : 118.39309
Path : inputFiles/topological_sort.c, Score : 101.57648
Process finished with exit code 0
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 353 ms (5 minutes ago)
1373:32 LF UTF-8 Git: master

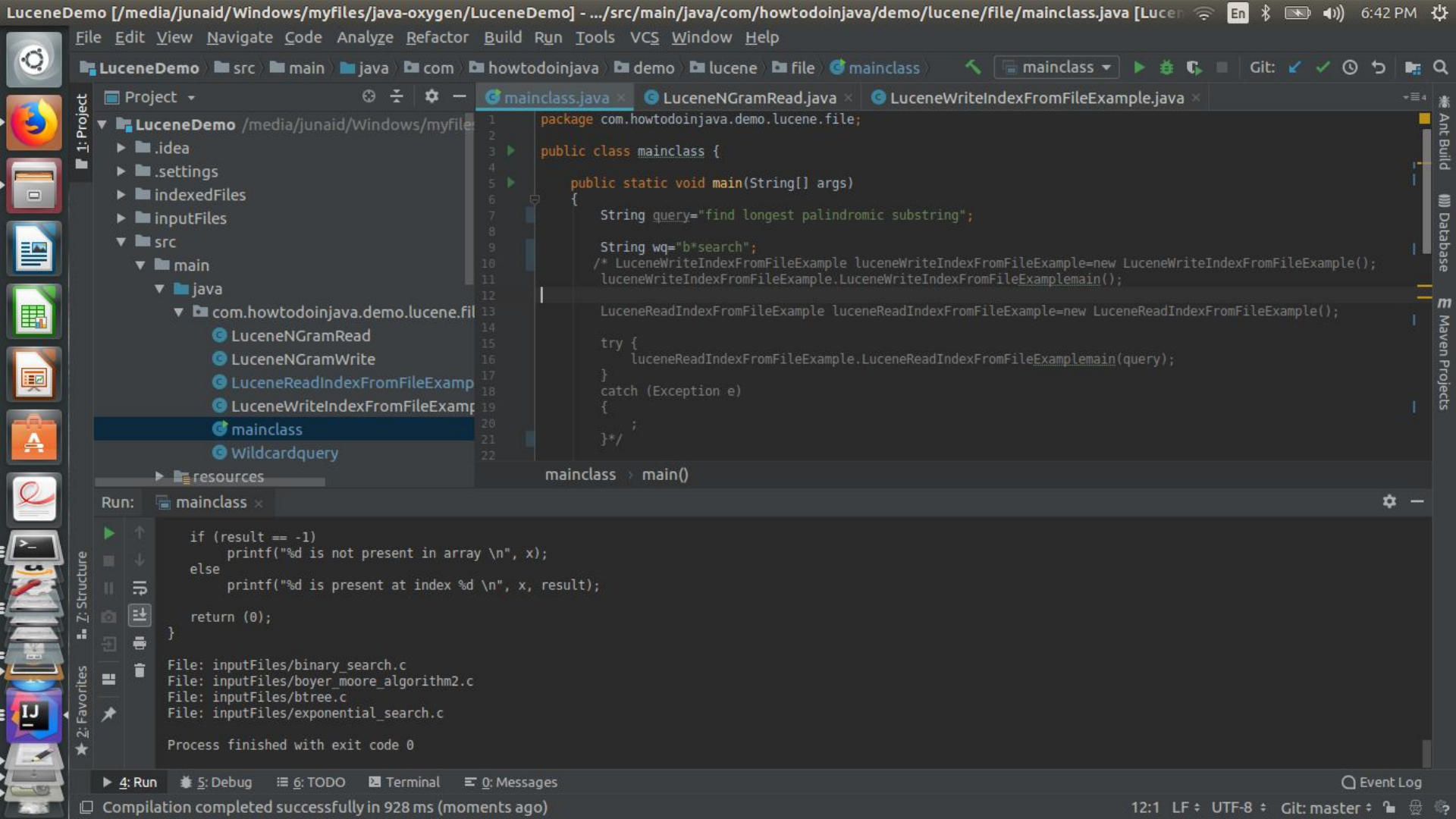
Query : Find longest common substring


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:37 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find longest common substring";
8
9 String wq="*search";
10 LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
mainclass › main()
Run: mainclass
inputFiles/union_intersection_in_list.c
inputFiles/validate_ip.c
inputFiles/validate_ipv4.c
inputFiles/validate_ipv6.c
inputFiles/vigenere_cipher.c
inputFiles/von_neumann_cellular_automata.c
inputFiles/warshalls.c
inputFiles/xor_swap.c
inputFiles/x_power_y.c
Total Results :: 249
Path : inputFiles/lexicographic_string_rank.c, Score : 228.78072
Path : inputFiles/permutations_of_string.c, Score : 177.76338
Path : inputFiles/std.c, Score : 124.09056
Path : inputFiles/heap_sort.c, Score : 114.00046
Path : inputFiles/matrix_chain_multiplication.c, Score : 112.78329
Path : inputFiles/validate_ip.c, Score : 111.880714
Path : inputFiles/infix_to_postfix.c, Score : 111.14948
Path : inputFiles/counting_sort.c, Score : 109.85071
Path : inputFiles/kruskal.c, Score : 104.55472
Path : inputFiles/postfix_evaluation.c, Score : 102.8937
Process finished with exit code 0
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 916 ms (moments ago)
6:6 LF UTF-8 Git: master
1: Project
7: Structure
2: Favorites
Ant Build
Database
Maven Projects

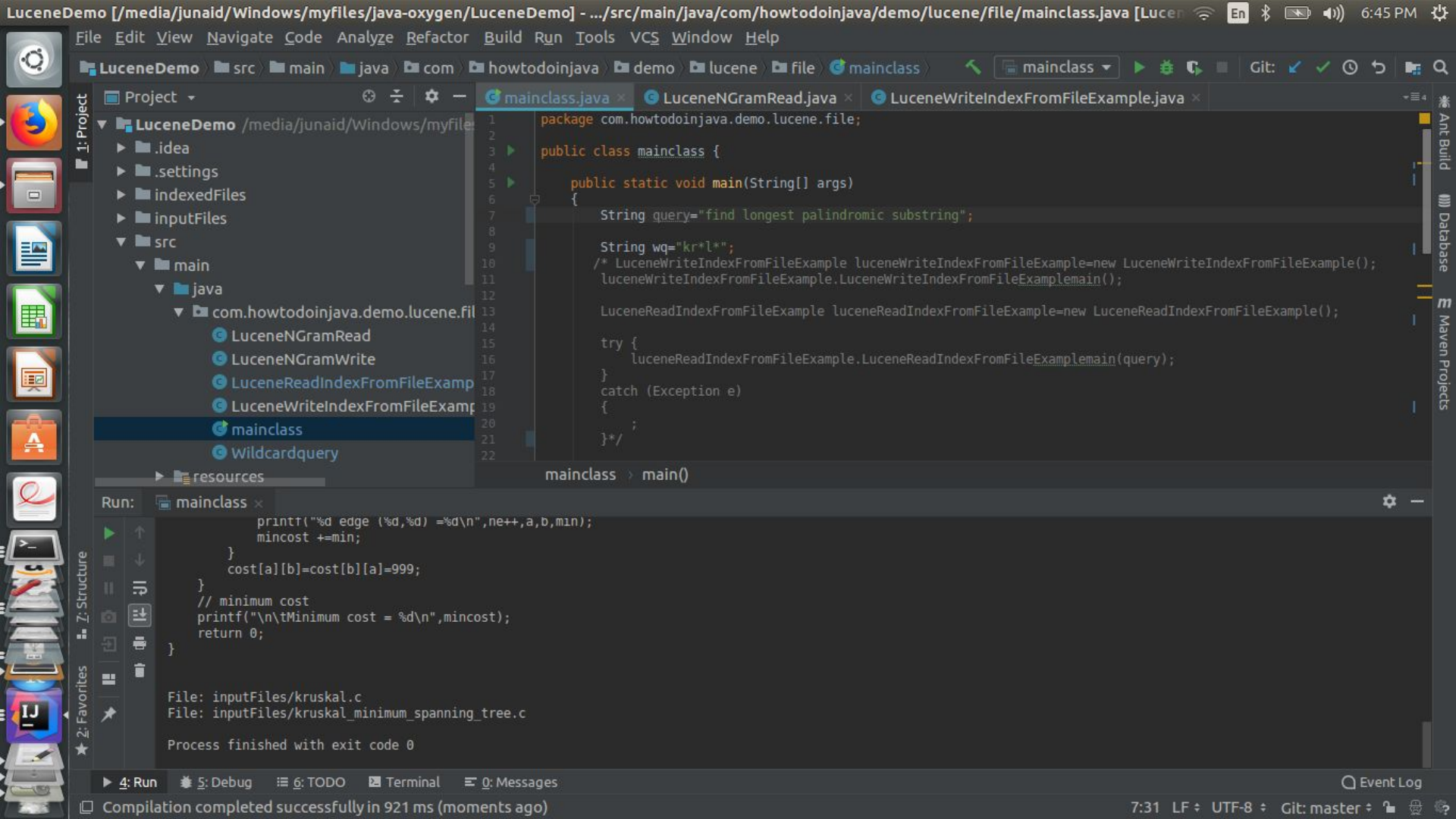
Query : Find the longest palindromic substring


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:39 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
1: Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find longest palindromic substring";
8
9 String wq="*search";
10 LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
mainclass › main()
Run: mainclass
i=parent[i];
return i;
}
int uni(int i,int j)
{
if(i!
Path : inputFiles/longest_palindromic_sequence.c, Score : 9.635367
Path : inputFiles/longest_bitonic_sequence.c, Score : 6.821668
Path : inputFiles/main.c, Score : 5.576691
Path : inputFiles/rabin-karp.c, Score : 5.458001
Path : inputFiles/biggest_suffix.c, Score : 5.0787907
Path : inputFiles/binary_search.c, Score : 4.0002604
Path : inputFiles/linear_search.c, Score : 3.8727293
Path : inputFiles/diameter.c, Score : 3.8550205
Path : inputFiles/ternary_search.c, Score : 3.777685
Path : inputFiles/kruskal_minimum_spanning_tree.c, Score : 3.715404
NGRAM INDEXING
inputFiles/2sum.c
inputFiles/addition.c
inputFiles/add_polynomials.c
Ant Build
Database
Maven Projects
7: Structure
2: Favorites
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 940 ms (moments ago)
868:1 LF UTF-8 Git: master

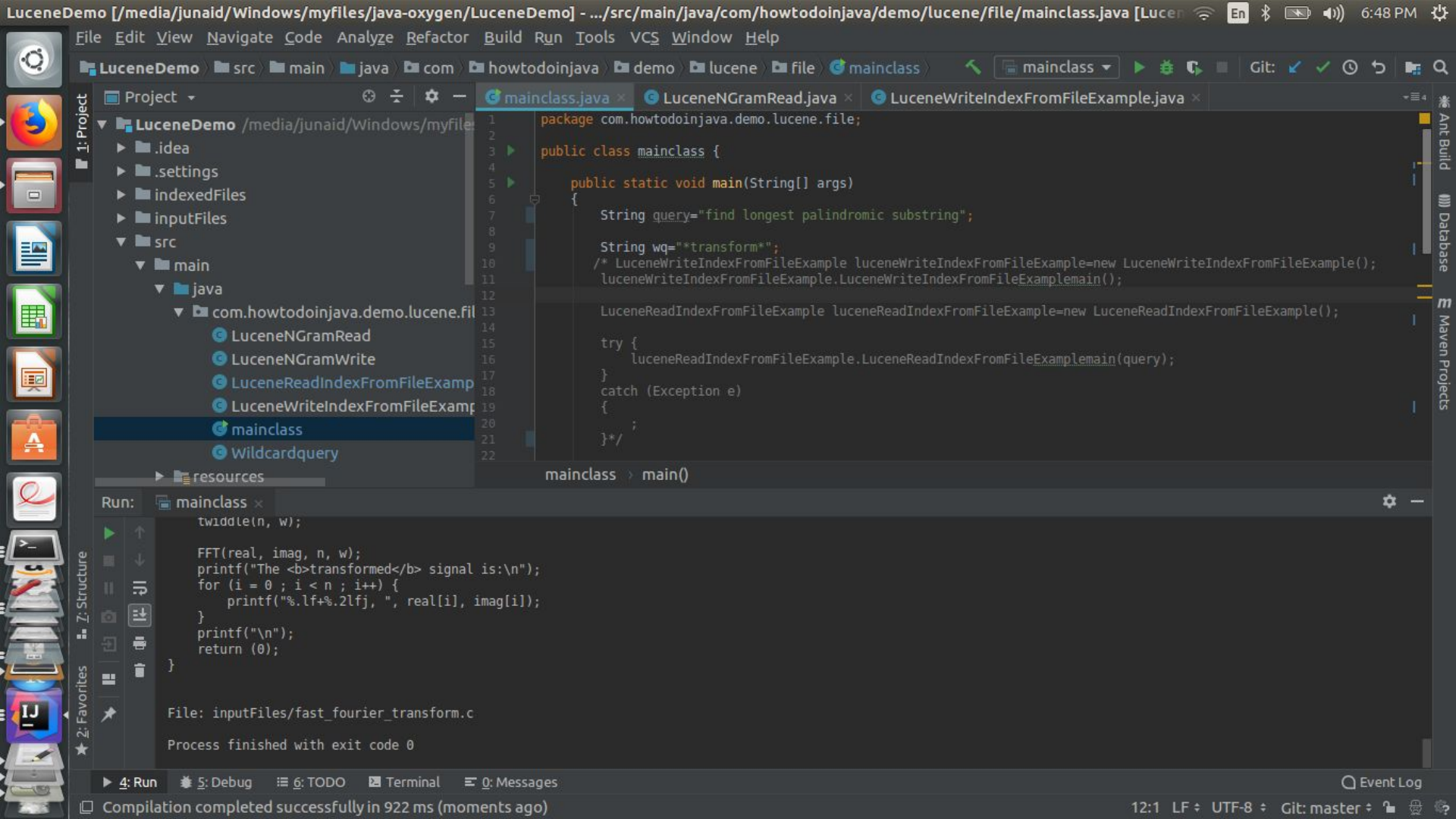
Query : b*search


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:42 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
1: Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
LuceneReadIndexFromFileExamp
LuceneWriteIndexFromFileExamp
mainclass
Wildcardquery
resources
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find longest palindromic substring";
8
9 String wq="b*search";
10 /* LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
15 try {
16 luceneReadIndexFromFileExample.LuceneReadIndexFromFileExamplemain(query);
17 }
18 catch (Exception e)
19 {
20 ;
21 }*/
22
mainclass › main()
Run: mainclass
if (result == -1)
printf("%d is not present in array \n", x);
else
printf("%d is present at index %d \n", x, result);
return (0);
}
File: inputFiles/binary_search.c
File: inputFiles/boyer_moore_algorithm2.c
File: inputFiles/btree.c
File: inputFiles/exponential_search.c
Process finished with exit code 0
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 928 ms (moments ago)
12:1 LF UTF-8 Git: master
Ant Build
Database
Maven Projects
7: Structure
2: Favorites

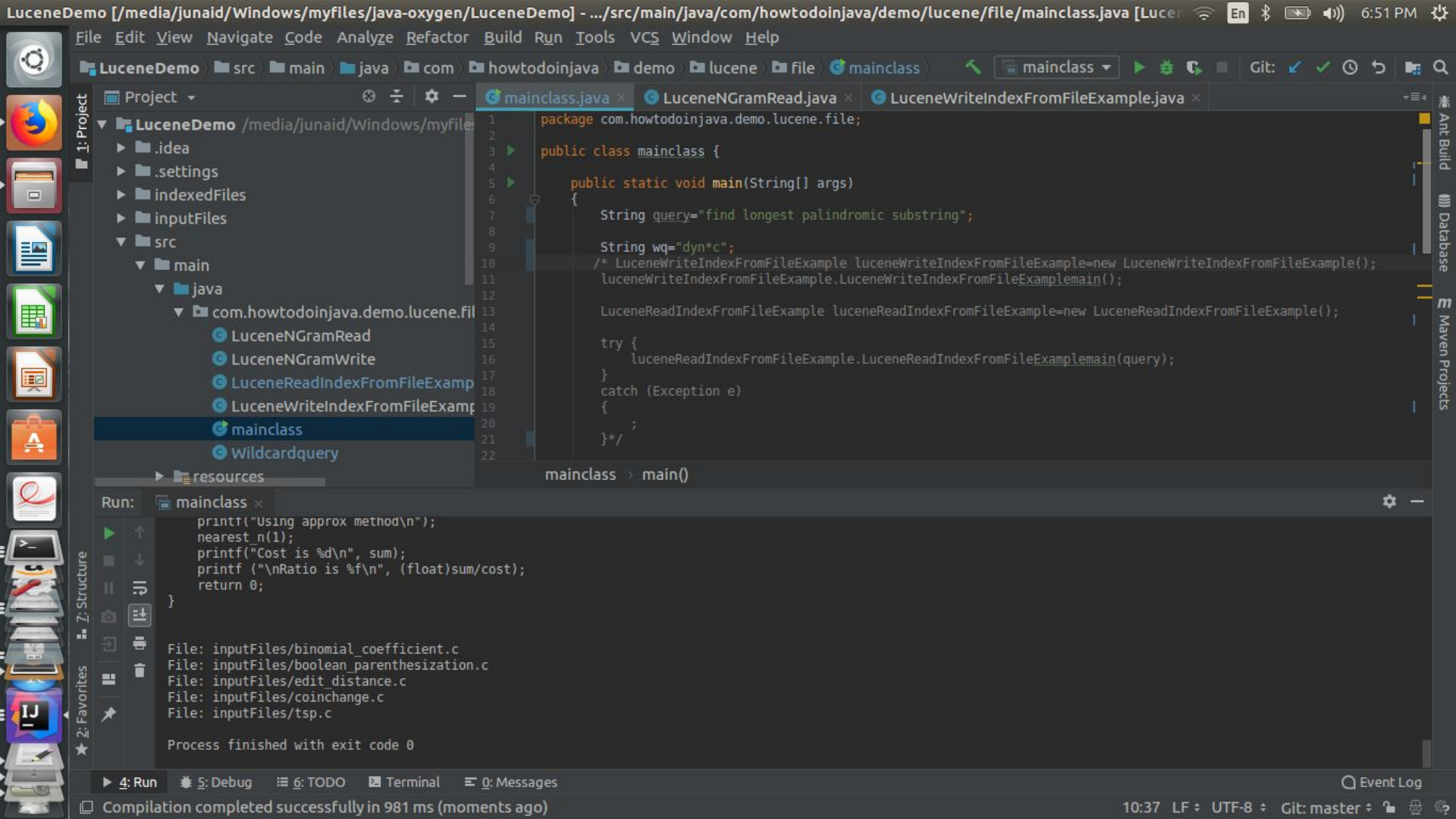
Query : kr*l*


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:45 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
LuceneReadIndexFromFileExamp
LuceneWriteIndexFromFileExamp
mainclass
Wildcardquery
resources
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find longest palindromic substring";
8
9 String wq="kr*l*";
10 /* LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
15 try {
16 luceneReadIndexFromFileExample.LuceneReadIndexFromFileExamplemain(query);
17 }
18 catch (Exception e)
19 {
20 ;
21 }*/
22
mainclass main()
Run: mainclass
printf("%d edge (%d,%d) =%d\n",ne++,a,b,min);
mincost +=min;
}
cost[a][b]=cost[b][a]=999;
}
// minimum cost
printf("\n\tMinimum cost = %d\n",mincost);
return 0;
}
File: inputFiles/kruskal.c
File: inputFiles/kruskal_minimum_spanning_tree.c
Process finished with exit code 0
4: Run 5: Debug 6: TODO Terminal 0: Messages
Event Log
Compilation completed successfully in 921 ms (moments ago)
7:31 LF UTF-8 Git: master
1: Project
7: Structure
2: Favorites
Ant Build
Database
Maven Projects

Query : *transform*


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:48 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
1: Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
LuceneReadIndexFromFileExamp
LuceneWriteIndexFromFileExamp
mainclass
Wildcardquery
resources
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find longest palindromic substring";
8
9 String wq="*transform*";
10 /* LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
15 try {
16 luceneReadIndexFromFileExample.LuceneReadIndexFromFileExamplemain(query);
17 }
18 catch (Exception e)
19 {
20 ;
21 }*/
22
mainclass › main()
Run: mainclass
twiddle(n, w);
FFT(real, imag, n, w);
printf("The transformed signal is:\n");
for (i = 0 ; i < n ; i++) {
printf("%.lf+%.2lfj, ", real[i], imag[i]);
}
printf("\n");
return (0);
}
File: inputFiles/fast_fourier_transform.c
Process finished with exit code 0
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 922 ms (moments ago)
12:1 LF UTF-8 Git: master
Ant Build
Database
Maven Projects
7: Structure
2: Favorites

Query : dyn*c


LuceneDemo [/media/junaid/Windows/myfiles/java-oxygen/LuceneDemo] - .../src/main/java/com/howtodoinjava/demo/lucene/file/mainclass.java [Lucen
6:51 PM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LuceneDemo src main java com howtodoinjava demo lucene file mainclass
mainclass
Git:
Project
1: Project
LuceneDemo /media/junaid/Windows/myfile
.idea
.settings
indexedFiles
inputFiles
src
main
java
com.howtodoinjava.demo.lucene.fil
LuceneNGramRead
LuceneNGramWrite
LuceneReadIndexFromFileExamp
LuceneWriteIndexFromFileExamp
mainclass
Wildcardquery
resources
mainclass.java
LuceneNGramRead.java
LuceneWriteIndexFromFileExample.java
1 package com.howtodoinjava.demo.lucene.file;
2
3 public class mainclass {
4
5 public static void main(String[] args)
6 {
7 String query="find longest palindromic substring";
8
9 String wq="dyn*c";
10 /* LuceneWriteIndexFromFileExample luceneWriteIndexFromFileExample=new LuceneWriteIndexFromFileExample();
11 luceneWriteIndexFromFileExample.LuceneWriteIndexFromFileExamplemain();
12
13 LuceneReadIndexFromFileExample luceneReadIndexFromFileExample=new LuceneReadIndexFromFileExample();
14
15 try {
16 luceneReadIndexFromFileExample.LuceneReadIndexFromFileExamplemain(query);
17 }
18 catch (Exception e)
19 {
20 ;
21 }*/
22
mainclass › main()
Ant Build
Database
Maven Projects
Run: mainclass
printf("Using approx method\n");
nearest_n(1);
printf("Cost is %d\n", sum);
printf ("\nRatio is %f\n", (float)sum/cost);
return 0;
}
File: inputFiles/binomial_coefficient.c
File: inputFiles/boolean_parenthesization.c
File: inputFiles/edit_distance.c
File: inputFiles/coinchange.c
File: inputFiles/tsp.c
Process finished with exit code 0
7: Structure
2: Favorites
4: Run
5: Debug
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 981 ms (moments ago)
10:37 LF UTF-8 Git: master

Conclusions

- K-gram returns a lot more results (not all are relevant)
- Wildcard queries are performing relatively better.
- Basic natural language queries give reasonable outputs.