



Project Explorer | knnquery.cc | knnquery.h | main.cc

- Non-Metric-Space-Library
 - Includes
 - CMakeFiles
 - debug
 - include
 - launches
 - sample.launch
 - lshkit
 - release
 - src
 - CMakeFiles
 - debug
 - factory
 - method
 - space
 - distcomp_bregmar
 - distcomp_js.cc
 - distcomp_lp.cc
 - distcomp_rankcorr
 - distcomp_scalar.cc
 - dummy_app.cc
 - experimentconf.cc
 - global.cc
 - knnquery.cc
 - logging.cc
 - main.cc
 - params.cc
 - query.cc
 - rangequery.cc
 - searchoracle.cc
 - tune_vptree.cc
 - utils.cc
 - CMakeCache.txt
 - CMakeLists.txt
 - Makefile
 - cmake_install.cmake

```

delete res;
}
}

```

Out | Ma

- stdio.h
- stdlib.h
- string.h
- time.h
- cmath
- limits
- string
- sstream
- vector
- fstream
- map
- global.h
- utils.h
- memory.h
- ztimer.h
- experiments.h
- experimentconf.h
- space.h
- spacefactory.h
- logging.h
- methodfactory.h
- meta_analysis.h
- params.h
- similarity
- multimap
- vector
- string
- stringstream

Properties for Non-Metric-Space-Library

type filter text

- Resource
- Builders
- C/C++ Build
 - Discovery options
 - Environment
 - Settings
 - Tool chain editor
 - Variables
- C/C++ General
 - Documentation
 - File Types
 - Indexer
 - Language Mappings
 - Paths and symbols
 - Project References
 - Refactoring History
 - Run/Debug Settings

Run/Debug Settings

This page allows you to manage launch configurations associated with the currently selected resource.

Launch configurations for 'Non-Metric-Space-Library':

- sample

Buttons: New... Duplicate Edit... Delete

Buttons: Restore Defaults Apply

Buttons: OK Cancel

```

MaxNumData,
MaxNumQuery,

<terminated> sample [C/C++ Local Application] gdb (2/5/14 10:45 PM)
locator<char> > > iterator:
typedef std::allocator<char> allocator_type;
typedef std::allocator<char>::reference reference;
typedef std::allocator<char>::const_reference const_reference;
typedef __gnu_cxx::__normal_iterator<char*, std::basic_string<char, std::char_traits<char>, std::allocator<char>> >
typedef __gnu_cxx::__normal_iterator<char const*, std::basic_string<char, std::char_traits<char>, std::allocator<char>> >
typedef std::reverse_iterator<__gnu_cxx::__normal_iterator<char const*, std::basic_string<char, std::char_traits<char>, std::allocator<char>> >

```