

Assignment 4: Road Map

Due date: Friday, December 20, 2019 at 22:00

This is an individual assignment. You may discuss it with others, but your code and documentation must be written on your own.

In a file called `roadmap.cpp` write a C++ library that manages a system of roads between cities in an area with islands. The library should use an internal representation of the roads to provide methods for adding roads and answering some basic queries:

- How many neighboring cities does each city have? Two cities are defined to be neighboring if they are directly connected with a single road.
- Is there a loop in the map? Starting from some city, is it possible to take a path through other cities, returning to the starting city, without traveling on the same road twice?
- Are two cities reachable? I.e. is there a path between two cities?
- How many islands are in the map? This is answered indirectly, by finding groups of cities that are isolated from each other.

The library must implement the `AbstractRoadMap` interface, as well as the `createRoadMap()` method, declared and documented in the `roadmap.h` header file attached below. In particular, you should create a new class that extends (inherits) from `AbstractRoadMap`. The example shown below illustrates the use of such a class and the semantics of its methods.

It is not important what you name this new class; what is important is that you implement `createRoadMap()` so that it returns a new instance of your road map class, initialized to be empty (i.e., should not contain any roads or cities).

Instructions

Submit one source file named `roadmap.cpp` through the iCorsi system.. Add comments to your code to explain sections of the code that might not be clear. You must also add comments at the beginning of the source file to properly acknowledge any and all external sources of information you may have used, including code, suggestions, and comments from other students. If your implementation has limitations and errors you are aware of (and were unable to fix), then list those as well in the initial comments.

To grade your implementation, we will run automated tests. Some such tests are provided with the on-line assignment package. However, notice that we may use other tests, and in any case that tests are necessarily incomplete, meaning that passing tests do not prove correctness. We will also evaluate your code for correctness and general quality independently of the automated tests.

You may use an integrated development environment (IDE) of your choice. However, *do not submit any IDE-specific file*, such as project description files. Also, *make absolutely sure that the file you submit can be compiled and tested with a simple invocation of the standard C compiler.*

Submit your solution through the iCorsi system.

roadmap.h

```
#ifndef __ROADMAP_H__
#define __ROADMAP_H__

#include <string>
#include <vector>

using std::vector;
using std::string;

class AbstractRoadMap {
public:
    /* Add a road that connects city1 with city2. All roads are bidirectional.
    */
    virtual void addRoad(const string & city1, const string & city2) = 0;

    /* Add a road that goes through all the cities in the cities
    * vector. In other words, add a road between each pair of
    * adjacent cities in the vector.
    */
    virtual void addRoad(const vector<string> & cities) = 0;

    /* Clear the road map completely.
    */
    virtual void clear() = 0;

    /* Return the number of neighboring cities for the given city, or
    * -1 if the given city is not in the map.
    */
    virtual int neighborCountForCity(const string & city) const = 0;

    /* Return true if and only if there is a path that starts from a
    * city X and then leads back to X by taking each no more than
    * once.
    */
    virtual bool hasLoop() const = 0;

    /* Return true if there is a path from city1 to city2.
    */
    virtual bool reachable(const string & city1, const string & city2) const = 0;

    /* Return the number of "islands" in the map. An island is a
    * maximal group of cities that are reachable from each other.
    * All cities in an island are reachable from each other, but they
    * are not reachable from any another city from another island.
    */
    virtual int countIslands() const = 0;

    /* Destructor
    */
    virtual ~AbstractRoadMap() {};
};

/* Return a new road map object that implements AbstractRoadMap. The
* new road map must be empty, meaning that it does not contain any
* road or city.
*/
extern AbstractRoadMap * createRoadMap();

#endif // __ROADMAP_H__
```

Example

Below is an example that adds some roads between cities and performs some queries.

```
#include <iostream>
#include <cassert>

#include "roadmap.h"

int main() {
    AbstractRoadMap * m = createRoadMap();

    // Add roads on one island:
    m->addRoad("Sengkol", "Sakra");
    m->addRoad({"Mataram", "Mangsit", "Pamenang"});
    m->addRoad({"Mataram", "Sengkol", "Koeta"});

    assert(m->countIslands() == 1);
    assert(m->reachable("Sengkol", "Mataram"));
    assert(m->neighborCountForCity("Mataram") == 2);
    assert(m->neighborCountForCity("Pamenang") == 1);
    assert(m->hasLoop() == false);

    // Create a loop:
    m->addRoad("Sakra", "Mataram");
    assert(m->hasLoop() == true);

    // Add roads on another island:
    m->addRoad({"Denpasar", "Sukawati", "Ubud"});
    assert(m->countIslands() == 2);

    // Cities on different islands are not reachable:
    assert(!m->reachable("Sengkol", "Sukawati"));

    delete(m);
    return 0;
}
```