BRIDGES Tutorial

Kalpathi Subramanian[1], Erik Saule, and Matthew Mcquaigue
krs@uncc.edu, esaule@uncc.edu, mmcquaig@uncc.edu

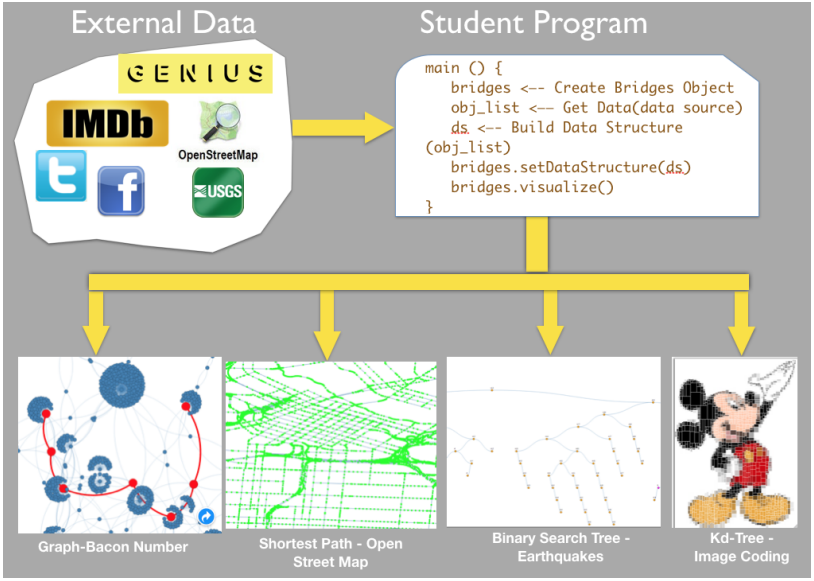
The University of North Carolina at Charlotte

BRIDGES Summer Workshop, May23-25, 2022

What is BRIDGES?

- An API to facilitate engaging assignments
- BRIDGES provides engaging input and output
- Easily incorporate real world datasets into routine class assignments that are more meaningful and span current interests of today's learners.
- BRIDGES can produce visualizations of student generated data structures, algorithm outputs/performance benchmarks, and support interactive games.
- BRIDGES provides the *building blocks for implementing data structures and algorithms, not their implementations!*

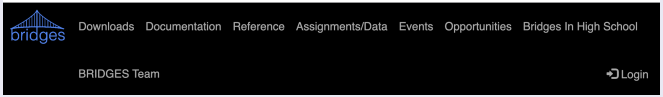
What is BRIDGES?

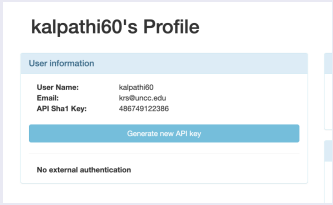


BRIDGES: Getting Started

BRIDGES Account, Credentials

- Go to the BRIDGES http://bridgesuncc.github.io/index.html and create an account by using the login button (you can use your email id as user name).
- Click on Profile button (in the upper right corner); you will see your credentials, user name, email and API Key; you will need this API key for every BRIDGES program you write.





- Every BRIDGES program will create the Bridges object and use the credentials as follows: *Bridges bridges = new Bridges(ASSIGNMENT_NUMBER, "USER_ID", "API_KEY")*

BRIDGES: Getting Started

BRIDGES Configuration/Installation

- **Java[JDK 8.0 and above]:**
  - Download the BRIDGES JAR file from the Downloads link on the Bridges website.
  - Augment your Java class path to include the path to the BRIDGES JAR file.
- **C++ [C++ 14 and above]:**
  - Download the BRIDGES C++ archive from the Downloads link on the Bridges website.
  - BRIDGES C++ uses the Curl library. This will need to be installed and BRIDGES programs need to be linked to the Curl library.
  - BRIDGES programs must be compiled with paths to the include and lib folders (the bridges library is only needed for the Game API).
- **Python [v 3.8 and above:]**
  - Use the following command to install the Bridges python sources:
    pip install bridges

BRIDGES: A Concrete Example

A BRIDGES Example Program: Linked List Using IMDB Actor Movie Data

```
int main() {
  //create the Bridges object, set credentials
  Bridges bridges(1, "BRIDGES_USER_ID", "BRIDGES_API_KEY");
  bridges.setTitle("Singly Liked List using IMDB Actor Movie Data");
  DataSource ds;
  std::vector< ActorMovieIMDB > am_list = ds.getActorMovieIMDBData(100);
  //building linked list
  SLelement<ActorMovieIMDB>* head = nullptr;
  for (auto im : am_list) {
    SLelement<ActorMovieIMDB>* am_node = new SLelement<ActorMovieIMDB> (
      im, im.getActor() + " - " + im.getMovie());
    am_node->setNext(head);
    // style nodes related to Cary Grant
    if (im.getActor() == "Cary_Grant") {
      am_node->setColor(Color("cyan"));
      am_node->setSize(30.);
      am_node->setShape(SQUARE);
    }
    head = am_node;
  }
```

BRIDGES: A Concrete Example

A BRIDGES Example Program: Linked List Using IMDB Actor Movie Data

```
    // style beginning node of list
    head->setColor(Color("red"));
    head->setSize(49.0);
    // send data structure to server, visuali
    bridges.setDataStructure(head);
    bridges.visualize();

    return 0;
}
```