CS-331 PROJECT 1: PATH FINDING

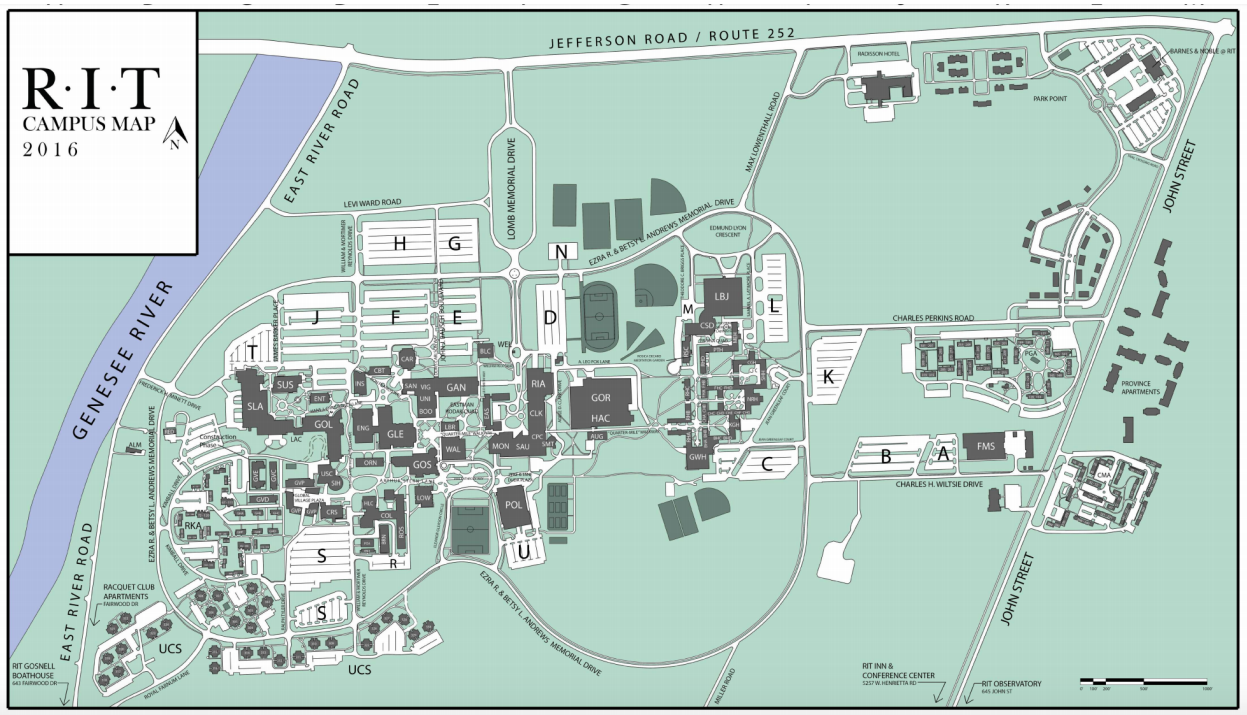
DUE DATE: 03/28/17 at 11:59 pm

100 points

You will solve the project in groups of two people. Submit one zip file per group with the code of the algorithms implemented, other needed files (i.e. filename.txt) and the solutions to the questions via Dropbox (through myCourses).

Problem:

For most of the Computer Science students, the Golisano building is the destination they want to reach from different parts of the RIT campus. This project is an activity though to apply different search strategies mentioned in the lecture to find paths from different corners of the RIT campus to the Golisano building. Here's the map of the RIT campus available in the website:



One representation of this map together with the path cost and the heuristic is given in the file **RIT_MAP.pdf**. The heuristic function is computed keeping the Golisano building as the final destination.

Objective: To apply some of the search strategies learned in class. Particularly, you must provide one implementation in JAVA for the algorithms mentioned below:

a) Breadth-First Search. **(20 points)**
b) Backtracking. **(20 points)**
c) A*. **(25 points)**

Requirements:

a) Data structure to store the information of the RIT map provided in the pdf called **RIT_MAP.pdf**. There are no restrictions on the data structure you want to choose.
b) For A*, one heuristic is already provided in the pdf attached. You need to come up with another heuristic and compare the performance of A* on both heuristics.
c) The algorithms will be run in the following way: **method_name filename.txt**, where filename.txt is a text file containing the information regarding the map. You need to create this file, so you can decide how to represent the information.
d) Each one of the algorithms will ask for a **start node** and a **finish node** (by using the standard input).
e) The output format :
    a. Output can be displayed on the command line or saved in a file.
    b. Paths must be given by showing each of the nodes which are part of that path together with the total cost obtained.

Write Up :

a) Give a high-level summary of how you implemented the algorithms required. What differences did you note in the performance of each one of them? **(10 points)**

b) If you gave to your code a map which was too large to fit in main memory, which would likely run faster, BFS, Backtracking or A*? **(5 points)**

c) In what ways are BFS and A* similar? **(5 points)**

d) Which search algorithms (if any) are guaranteed to find the shortest path to the goal? What does it depend on? **(5 points)**

e) According to the results obtained by the different algorithms implemented, how do they perform when compared to the way humans would do in the real world (not necessarily moving from building to building)? **(5 points)**

f) How would you define the same problem in order to represent the way humans would move through the map? **(5 points)**