Location Based Assignments in Early CS Courses Using *BRIDGES* Engages Students[*]

Matthew Mcquaigue[1] Jay Strahler[1] Kalpathi Subramanian[1]
Erik Saule[1] Jamie Payton[2]
[1]Computer Science, The University of North Carolina, USA
[2]Computer Science, Temple University, USA
{mmcquaig, jstrahl1, krs, esaule}@uncc.edu, payton@temple.edu

Abstract

Freshmen and sophomore level courses in computer science are critical to long-term student development and success. At the same time, these courses, such as data structures and algorithms are usually challenging and require significant motivation to keep students engaged. In this work, we present through our *BRIDGES* system a set of *location* based assignments that can serve to reinforce core concepts and algorithms by placing them in more meaningful settings and applications, and demonstrate the relevance of computing in the early stages of a student's career. We performed a small pilot study using a subset of these assignments in a special topics course on algorithms, and conducted student surveys after each assignment. The surveys were unanimously positive, and the students enjoyed coding the algorithms as well as the datasets and visualizations associated with the assignments.

1 Introduction

While enrollments in computer science have grown significantly in recent years, future trends indicate a significant need for CS graduates to join the modern workforce [5]. Future jobs will increasingly be more technical, especially in areas such as AI, vision, machine learning, data science and security. Thus,

[*]Copyright ©2021 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

it is imperative that CS majors possess a strong foundation in their early years to ensure their long-term success; however, foundational courses can be rigorous and not very engaging, and the field has seen significant attrition, especially in the freshmen/sophomore years, and for groups traditionally underrepresented in computing. Engaging majors in the early stages of their CS program with rigorous, yet meaningful and engaging content can provide the means to demonstrate the relevance and possibilities of computing, and help retention.

As part of the *BRIDGES* system [6, 4], we have developed a repository of assignments for early courses in computer science [3]. We had also presented how these could be used in CS1/CS2 [2] and algorithm analysis courses [21]. The *BRIDGES* API (available in Java, C++ and Python) facilitates the use of real-world datasets in early CS courses and uses visualizations to demonstrate student generated work, while maintaining course rigor.

In this work we present a class of *location based assignments* that we have recently developed as part of the *BRIDGES* assignment repository; these assignments rely on data sources that have location (Lat/Long) information. These assignments have two advantages, (1) they continue to provide highly engaging assignments with visual output of classic CS concepts/algorithms, and (2) students' output can be customized to locations or concepts of their own interest. This gives more flexibility to assignments, and getting away from the 'one size fits all' characteristic of typical course assignments. We report on a pilot study of using these assignments in an algorithms course.

2 Related Work

Engagement Strategies. In recent years, a significant amount of effort has focused on strategies to engage and motivate students in early CS courses, in an effort to retain CS majors. Engagement can span many dimensions, involving skills, participation/interaction, emotional and/or performance [11]. Engagement strategies can be classified as either *content-based* or based on pedagogical activities. Content based engagement focuses on making the content of the course or associated activities (assignments, lectures, videos, etc) meaningful and relevant. Engagement strategies based on activities include those based on active learning techniques, which in turn could be based on any combination of lab-based instruction, flipped classroom settings, gamification, peer-learning, and use of multimedia content [18, 10, 12].

Learning Materials and Repositories. Repositories containing learning materials (slides, assignments, videos) are another mechanism to assist instructors find engaging content; these include Nifty Assignment collections [17], EngageCSEdu [13], ModelAI [1]. Some of these tend to include the ‘fun’ factor, such as those based on games [22].

Location Based Materials Many of the Nifty assignments are location based assignments, such as the N-body simulation [20], star chart display [19], voting districts and gerrymandering [25, 15], that lend themselves to nice visualizations that can be highly engaging. Using games and AI as part of assignments also lend themselves to using geometric spaces [7].

3 Location based Datasets

3.1 Earthquakes - US Geological Survey (USGS)

USGS records earthquake data [23] in real-time and makes it available in real time. We have built an interface to gather these earthquake records periodically and store the 10,000 most recent earthquakes in a database. Our *BRIDGES* system provides an API call to retrieve a desired number of these records as objects in C++, Java or Python. Each earthquake record contains the quake magnitude, location (lat/long and quake area), and epoch time.

3.2 OpenStreet Map - Street Networks, Amenity Data

OpenStreet Maps is a publicly available resource [16] providing street networks at multiple resolutions, as well as amenities (schools, hospitals, restaurants, fire stations, etc.). This data, while extremely rich, is also based on contributions by the public and does not guarantee uniform coverage of all areas.

We downloaded the entire map of North America from GeoFABRIK [8]. Additional processing included removing extraneous data to save storage. The data was held in 2 parts, one for the road and street data, and the second holds amenity data, such as restaurants, schools, etc.

In *BRIDGES*, we can retrieve the OSM street network data by specifying a Lat/Long bounding box; the system then returns the set of vertices and edges that make up the street network. Each vertex contains an id and lat/long information. Edges are specified by the source and destination vertex ids. Amenity data is retrieved through a separate query, by providing the amenity name within a rectangular Lat/Long region.

3.3 City/Country Data

The city/country data was obtained from GeoNames [9]. This site provides a single file with every city in the world. Currently, we use only cities in the United States. Each city instance is part of a collection that has a schema containing an id, city, state, country, latitude, longitude, elevation, and population, and time zone attribute. Currently, *BRIDGES* allows querying this data using a combination of the above attributes; alternately, a Lat/Long bounding box can be specified to extract all the cities contained within the box.

3.4 Elevation Maps and Navigation

Elevation model data was obtained from NOAA [14] on per query basis, using a Lat/Long bounding box. The data is saved locally to the server for future requests. Utility tools are used to convert the raster data into a more human understandable form for use by end users. Note that the resolution varies depending on the queried region, and is specified as part of the query.

3.5 US Census

US Census data, published every 10 years, is available from the US Census bureau [24]. The data collects data on states, counties, census tracts, block groups and blocks. Basic demographic data such as race and sex are part of the data.[1]

4 Location Based Problems, Assignments, Applications

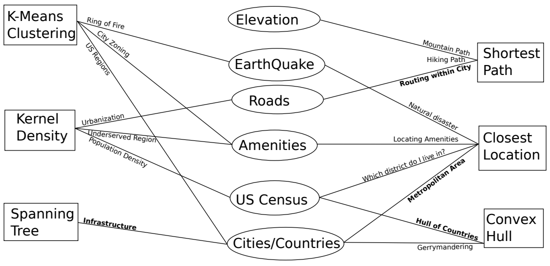

Figure 1: Location Based Assignments: Left and Right columns illustrate relevant algorithms that can take advantage of data sources that contain location information. Middle column illustrates examples of data sets that we have used to construct programming assignments using BRIDGES, all of which lend themselves to visual outputs. Edge labels indicate potential applications.

[1]we have only recently begun looking at this dataset for use with *BRIDGES*.

4.1 Kernel Density

In the kernel density problem, a space (here a 2D space) is discretized in cells. And there are points that may carry a weight. In the density problem, one counts for each cell the (weighted) number of points in the surrounding of that cell. One can count the number of points within the cell or in proximity of the cell. These maps are common in geography. Using different data lead to different analyses, while being an equivalent assignment from a learning objective standpoint:

Population Density. The simplest application of Kernel Density is to compute population density maps using US Census data. One can also use ethnical attribute or age attributes to get a finer picture of US demographics. The census bureau also provide voter registration data which can help further understand voting patterns in the US such as the urban/rural divide.

Underserved region. Using Amenity data collected by Open Street Map, one can compute density maps for public and private services, such as schools, hospitals, restaurants, grocery stores. This can enable identifying food deserts or communities underserved by local governments.

Urbanization. Using road data from Open Street Map, kernel density will provide maps of urbanization of a region.

4.2 Shortest Path

A classic assignment when using map based data is to compute shortest path. There are a number of shortest path algorithms such as Dijkstra's and Ford Bellman, that are applicable. Shortest Path is also appropriate to discuss artificial intelligence methods with algorithms like A*.

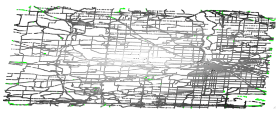

Figure 2: Dijkstra's shortest path algorithm on OpenStreeMap data of Minneapolis

Routing within City. GPS routing is a feature students use commonly and they can get the chance to implement their own. One can use the Open Street Map street level data to build a graph and compute shortest path on it. *BRIDGES* visual attributes can be used to show distance variation from the source, as can be see in Fig. 2.

Mountain Path. The mountain path assignment as written in the Nifty assignment collection [17] works from an elevation map (a gray scale image) which *BRIDGES* makes available from NOAA. The task is to find the lowest cost path, from the left end of the image to the right end of the image, where the cost is defined as the sum of difference in elevation between consecutive pixels in the path. The Nifty assignment uses a simple greedy approach to find a path which is suitable in various classes from CS1 to Algorithms. A visualization of such a path can be seen in Fig. 3. One can find the optimal path that only ever goes right using a dynamic programming algorithm which is suitable for an Algorithms course. One can also model the image as a graph where each pixel has 8 neighbors and use a shortest path algorithm.

Biking/Hiking Paths. One can combine both Open Street Map street level data and elevation data from NOAA to create a graph for cycling and hiking paths. The length of edges in the graph can be adjusted to account for the difficulty of going uphill and the ease of going downhill.

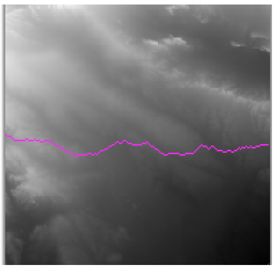
Figure 3: Mountain Path

4.3 K-Means Clustering

K-means clustering is a simple unsupervised method to find clusters in a spatial dataset. A number cluster centroids are distributed randomly initially. Each point in the database is allocated to the region represented by the closest centroid. Then the centroid is updated to be at the barycenter of the points in the region it represents. While k-means clustering is refined further in data mining courses, the naive scheme is suitable for implementation by CS1 students.

Ring of Fire. Computing K-means cluster on the location of recent earthquakes will highlight geological properties of the Earth; For instance, many earthquakes are located around the Pacific ocean and is known as the Ring of Fire.

US regions. Computing K-means on the US city map will highlight the structure of the country in a few main region. And it will highlight a known east coast/west coast divide.

US city zoning. Clustering commercial amenities (from OpenStreetMap) of a particular type will identify commercial zoning in most US city. Alternately, clustering the population of a city using census data will identify residential zoning. The two maps can be compared.

4.4 Closest Location Queries

Given a set of points in a (here 2D) space, a simple query is to find the closest point in the set to a particular location. This form of spatial query is suitable for different levels. At a CS1 level, one can simply put the set in a linear structure, such as a list or an array, and iterate over it to find the closest match. One could also use a uniform partition of the space to accelerate the search. This can be suitable for a project in CS1/CS2. At data structure level, one could do adaptive partitioning such as quad-trees and K-D trees. Students could even implement different strategies and compare their performance.

Locating Amenities. With OpenStreetMaps data, one can answer questions such as 'Where is the restaurant that is closest to me?', or in a GPS routing scenario, "what is the street corner closest to my location?"

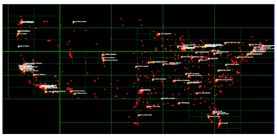
Figure 4: Quadtree Partitioning of 1000 cities

Natural Disaster Risk. With Earthquake data, one can compute for a particular location the closest large earthquake that has been reported.

Identifying metropolitan areas. Using the city location data, the assignment could aim to identify the closest city of a particular size of a particular location. Fig. 4 illustrates an example quadtree partitioning on a set of 1000 cities. This can be further used to approximate Voronoi decomposition of the country based on the 1000 largest cities.

Which census district do I live in? While census districting can be complicated, a student can use location query to identify the closest census district their home is part of and look at the demographics of that district.

4.5 Convex Hull

Location data can also enable a range of geometric problems. A classic one is to compute the convex hull of a set of points. This is useful for visualization purposes but also as a preprocessing step to compute collisions in a physics engine. Computing the convex hull can be done with a range of methods; Jarvis March is a brute force algorithm suitable for CS1; Divide and Conquer (similar in its analysis to merge sort) and Quick Hull (similar to quick sort) are suitable for algorithm courses.

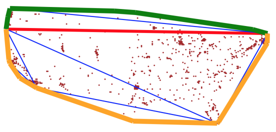
Figure 5: Convex Hull using divide and conquer alg. on US city data

Convex Hull of countries and states. One can use the City data to use as a the input to a convex hull algorithm. This enables the student to see how much their country or state looks like their convex hull. A convex hull of US cities can be seen in Fig. 5.

Visualizing gerrymander. Convex Hull can be used on congressional voting districts (extracted from US census data) as an approximation of gerrymandering.

4.6 Minimum Spanning Tree

Minimum Spanning Trees (MST) are classic chapters in algorithm books and courses with two primary greedy algorithms: Prim's and Kruskal's algorithm. Minimum Spanning tree is often a topic in Algorithms courses. A variant of spanning trees are Steiner Trees that are often used to plan infrastructure project such as train lines, power lines, water lines, optic fibers. However, Steiner trees are often not presented, as the problem is more complicated and does not have an optimal polynomial algorithm.

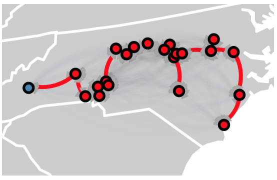
Figure 6: Prim's MST alg. on cities in North Carolina

Minimum Spanning Tree using City Data. Computing Minimum Spanning tree of city/country data serves as a approximation of infrastructure. Fig. 6 illustrates the MST of a set of cities in North Carolina.

5 Pilot Study: Student Reflections/Feedback

In the fall 2021, a special topics course entitled 'Real World Algorithms' was offered to Junior/Senior CS students at UNC Charlotte for the first time. The course required Data Structures as a prerequisite[2]. The course is being considered for a more permanent offering as part of our degree concentration. Four of the location based projects detailed in Section 4 and highlighted in Fig. 1 were included. The course had five students enrolled. Students completed a project/reflection survey after each module.

Students completed 3 project surveys in the course. The location based projects in the course were part of 2 of the surveys. In particular, it had 2 free form questions:

- Identify the essential concept(s) learned by completing the assignment, and indicate why you think that concept may be important to understanding and learning how to program.
- Comment on what aspects of the project were interesting, meaningful, engaging (or not)? Also, indicate why you liked or disliked using *BRIDGES*.

The surveys were unanimously positive. Students enjoyed coding the algorithms like MST, Convex Hull and Dijkstra's shortest path algorithms as well as the datasets used with the algorithm ('learning the functionality...how to implement Djikstra's algorithms...further increased my confidence in programming', 'greater understanding of geometric structures...how to manipulate them as a data structure'). All students commented positively on the visualization

[2]Our current program does not contain a separate algorithm analysis course

capabilities of *BRIDGES* ('enjoyed graphically displaying the Convex Hull', 'served as a robust visualization and debugging tool') as it helped them understand the output of these algorithms, which are naturally suited to be visually salient.

6 Conclusions

In this work, we presented a class of location based assignments that are highly engaging, visual and can be adopted across the freshmen and sophomore CS courses such as CS1, CS2, Data Structures and Algorithm Analysis. These assignments provide a vital connection to real world applications and problems that will be attractive to CS majors. More importantly, they provide the key advantage of making it *fit the interests of students by making it possible to explore such applications to the location or region of their interest.* We report preliminary results from an algorithm analysis course that tied classic computer science algorithms to real world applications. Specifically, it focused on coupling traditional algorithms to location based datasets, so as make the course content more meaningful to CS students. A small pilot study of an algorithms course was presented where some of these algorithms were assigned as part of the course. Feedback from students were unanimously positive, in terms of both the course content as well as the use of the *BRIDGES* system, which facilitated such real world content and visualization capabilities. However, this study had only five students and was not a required course, thus, results will need to be replicated in both a larger course and avoid self-selection bias. We plan to make this course a more regular offering as part of a degree concentration in the coming year and gather and analyze more formal feedback.

7 Acknowledgements

This work was supported in part by grants from the National Science Foundation (Nos. 1726809, 2142389).

References

[1] AAAI. Model AI Assignments, 2018.

[2] Allie Beckman, Matthew Mcquaigue, Alec Goncharow, David Burlinson, Kalpathi Subramanian, Erik Saule, and Jamie Payton. Engaging Early Programming Students with Modern Assignments Using BRIDGES. volume 35, page 74–83, Evansville, IN, USA, apr 2020. Consortium for Computing Sciences in Colleges.

[3] BRIDGES Development Team. BRIDGES Assignment Repository. http://bridgesuncc.github.io/newassignments.html, 2022.

[4] BRIDGES Development Team. BRIDGES Website. http://bridgesuncc.github.io, 2022.

[5] Bureau of Labor Statistics. Occupational outlook handbook. https://www.bls.gov/ooh/computer-and-information-technology/computer-and-information-research-scientists.htm#tab-6.

[6] David Burlinson, Mihai Mehedint, Chris Grafer, Kalpathi Subramanian, Jamie Payton, Paula Goolkasian, Michael Youngblood, and Robert Kosara. BRIDGES: A System to Enable Creation of Engaging Data Structures Assignments with Real-World Data and Visualizations. In *Proceedings of ACM SIGCSE 2016*, pages 18–23, 2016.

[7] Adrian A. de Freitas and Troy B. Weingart. I'm going to learn what?!? teaching artificial intelligence to freshmen in an introductory computer science course. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, page 198–204, New York, NY, USA, 2021. Association for Computing Machinery.

[8] GeoFABRIK. Openstreetmap data extracts. https://download.geofabrik.de/.

[9] GeoNames. Geonames geographical database. http://www.geonames.org/.

[10] Mark Guzdial. A media computation course for non-majors. In *Proceedings of the ITICSE 2003*, pages 104–108, 2003.

[11] M.M. Hendelsman, W.L. Briggs, N. Sullivan, and A. Towler. A measure of college student course engagement. *Journal of Educational Research*, 98(3):166–175, 2005.

[12] Diane Horton, Michelle Craig, Jennifer Campbell, Paul Gries, and Daniel Zingaro. Comparing outcomes in inverted and traditional CS1. In *Proceedings of the ITICSE 2014*, pages 261–266, 2014.

[13] Alvaro Monge, Beth A. Quinn, and Cameron L. Fadjo. Engagecsedu: CS1 and CS2 materials for engaging and retaining undergraduate cs students. In *Proceedings of ACM SIGCSE*, SIGCSE '15, pages 271–271, 2015.

[14] howpublished = https://www.ngdc.noaa.gov/mgg/global/global.html NOAA, title = Digital Elevation Models.

[15] A. Obourn. Gerrymandering. http://nifty.stanford.edu/2019/obourn-gerrymandering/.

[16] OpenStreetMap contributors. Openstreemap. https://www.openstreetmap.org, 2021.

[17] Nick Parlante. Nifty assignments, 2018.

[18] Johanna Pirker, Maria Riffnaller-Schiefer, and Christian Gütl. Motivational active learning: Engaging university students in computer science education. In *Proceedings of ITICSE*, pages 297–302, 2014.

[19] K. Reid. Star charts and constellations. http://nifty.stanford.edu/2009/reid-starmap/.

[20] R. Sedgewick and K. Wayne. N-body simulation. http://nifty.stanford.edu/2017/wayne-nbody/.

[21] Jason Strahler, Matthew Mcquaigue, Alec Goncharow, David Burlinson, Kalpathi Subramanian, Erik Saule, and Jamie Payton. Real-world assignments at scale to reinforce the importance of algorithms and complexity. *J. Comput. Sci. Coll.*, 35(8):166–175, apr 2020.

[22] Kelvin Sung, Rebecca Rosenberg, Michael Panitz, and Ruth Anderson. Assessing game-themed programming assignments for CS1/2 courses. In *Proceedings of GDCSE*, GDCSE '08, pages 51–55, 2008.

[23] US Geological Survey. USGS earthquakes.

[24] United States Census Bureau. 2020 census state redistricting data (public law 94-171) summary file technical documentation. Technical report, August 2021.

[25] K. Wayne. Purple america. http://nifty.stanford.edu/2014/wayne-purple-america/.