

ABOUT: I work for the Admissions Office as a Diversity Outreach Intern. One of the sub-teams I have worked on is referred to as the “hosting team.” This team matches visiting students to an Amherst student based on similarities. This project is a start to creating a program that would read in the survey results and match hosts to students! It currently matches based on “prospective major” and “major” and prints the results.

DATA: I have created text files titled “preFresh” and “host.” Each string “name” MUST be followed by two other strings: (major and room (or N/A)). In reality, the “hosting team” uses data acquired by survey results compiled into an excel sheet.

DATA STRUCTURES: Data Structure #1 = Hash Table Data Structure each Person object is added to a different index in the Hash Table based on the hash function that correlates to specific majors. Each List in the Hash Table is accessed by using a “delete” method that removes from the lists and pairs the preFresh to the Host. Data Structure #2 = Adjacency List each Person object is added to a different vertex of the graph based. Each vertex represents a different major in this program and ultimately acts as a sorting feature! The preFresh and Hosts are matched using the same function as the Hash Table. The utilization of both of these structures can be seen in main where I have labeled them. Additionally the output will notate which structure is being used. With the way I have implemented the graph data structure as an adjacency list it follows a very similar rule to a hash table. As a result, I believe there is no large difference between big-O time complexity for either data structure. However, I would use a graph over hash table for future projects as we could possibly run into duplicates.

INSTRUCTIONS: compile all files in the folder (javac *.java), run “java Main” in the terminal to begin pairing. Text files MUST follow the structure of: “name” return “major/ prospective

major” return “room/ n/a” this is necessary for the information to be properly read by the program and placed into the Person object arrays.

FUTURE: In reality, the “hosting team” matches are based on much more than just majors. I hope to further the matching capabilities of my program so it takes into account gender + gender preference, interests, and affinity. I could do this based on a hierarchy of importance. Students would rank which of the categories are most important to their open house experience and will be matched to a host based on importance!