# YOUR PROJECT TITLE AND ACRONYM REPLACE THIS

# YOUR NAME REPLACES THIS

# Assignment 4 1/31/2023

## 1 SUMMARY DESCRIPTION—UPDATED (OR REPEATED IF UNCHANGED)

For this project, I want to do a classification based on the destination of the package, and classify them according to the city, state and country where the package is located. So the first choice is not to seek their common ground. This part can be done using divide and conquer. We will talk about the international classification of the package according to the city classification of the package.

In this assignment, based on the automatic classification of packages, arranged a priority for each package, and then told the user the priority of package delivery from previous assignment. I made some changes. I use the Dijkstra algorithm to let code decide which package has higher priority based on their destination. The default city is Boston. Much closer to the Boston, much higher priority.

Your response should replace this.

## 2 I/O EXAMPLE FROM PROJECTED COMPLETED PROJECT—UPDATED (OR REPEATED IF UNCHANGED)

Input csv file example:

1, MA, Worcester

2, AZ, Tucson

3, MA, Boston

4, AZ, Phoenix

5, CA, San Francisco

6, CA, San Diego

7, NV, Las Vegas

Output:

Prioritization:

Package: 3, Package: 1, Package: 2, Package: 4, Package: 5, Package: 6, Package: 7

## 3 REQUIREMENTS IMPLEMENTED IN THIS RELEASE

### 3.1 Input requirement

### Input should contain package number,state,city.Because i wanna use the algorithm to calculate which city has shortest path and make it has higher priority, so this this not require input priority.And in the input form should looks like the test data I provide above.

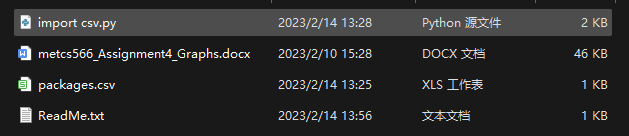
### 3.2 Output requirement

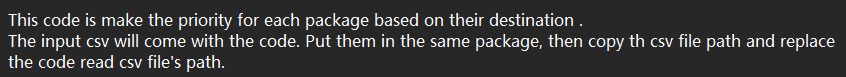
Output should contain package number and put them in order by priority from highest to lowest.

More requirements if applicable

## 4 ILLUSTRATIVE OUTPUT

## 5 YOUR DIRECTORY



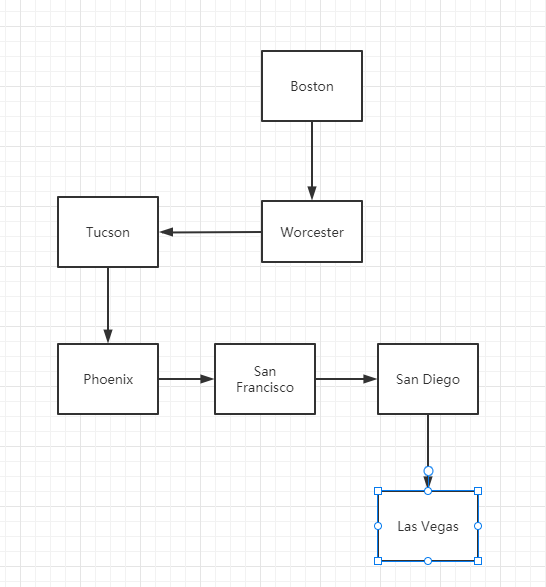


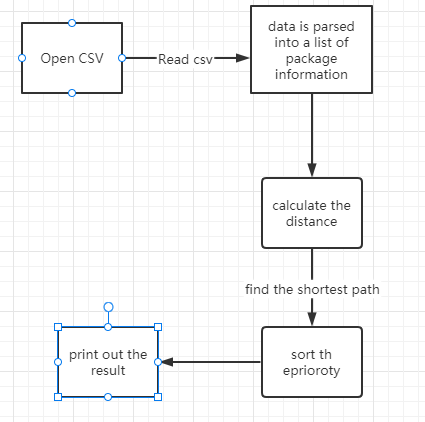
## 6 GRAPH PROCESSING IMPLEMENTED

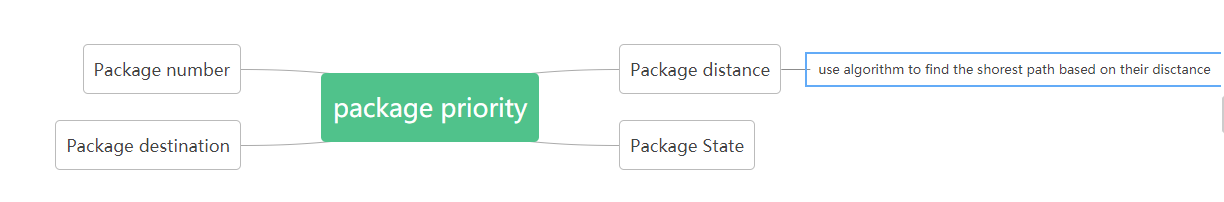
In this project, I used a number of different cities to let the program automatically prioritize the packages. Due to possible duplicate routes. In order to guarantee the shortest path, I used Dijkstra algorithm to find the shortest path. This ensures that the optimal priority is found after walking the least repeated path

### 6.1 Class model and Sequence Diagram

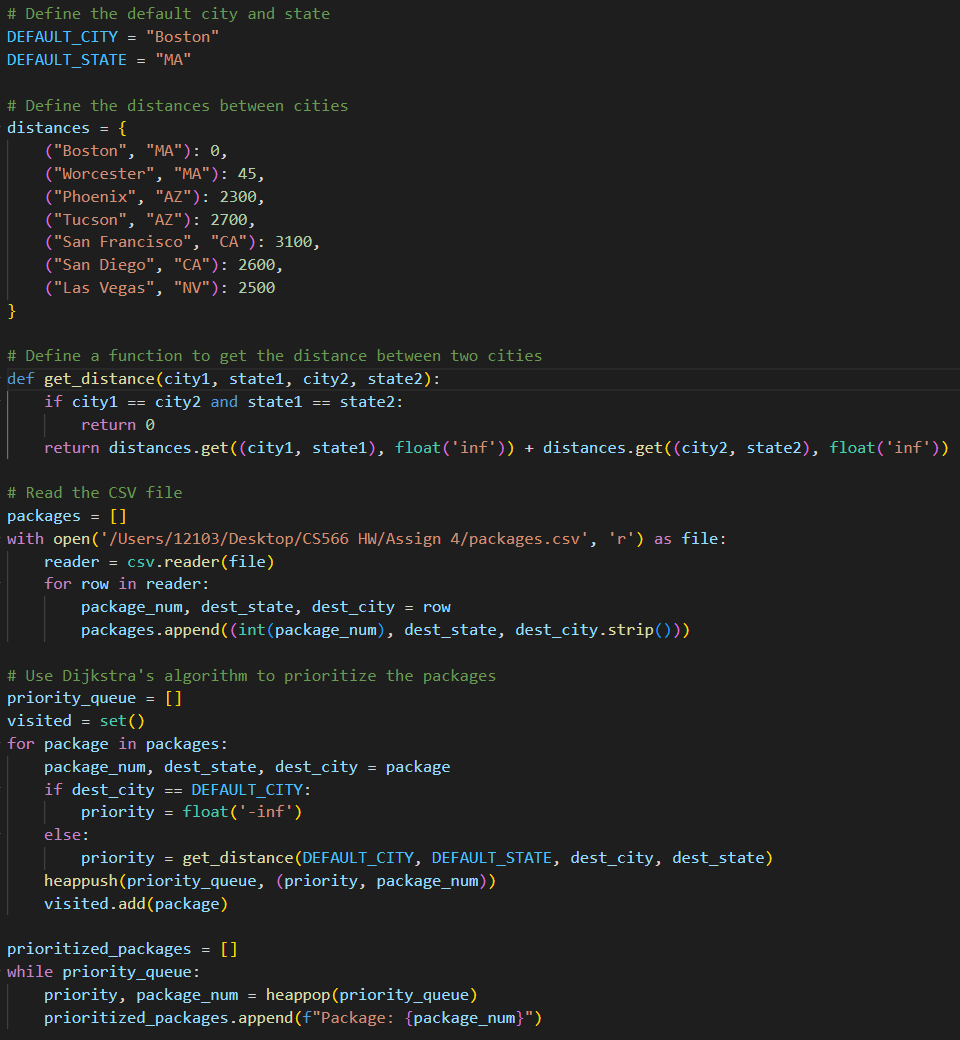
### Path Graph







### 6.2 Code showing graph processing



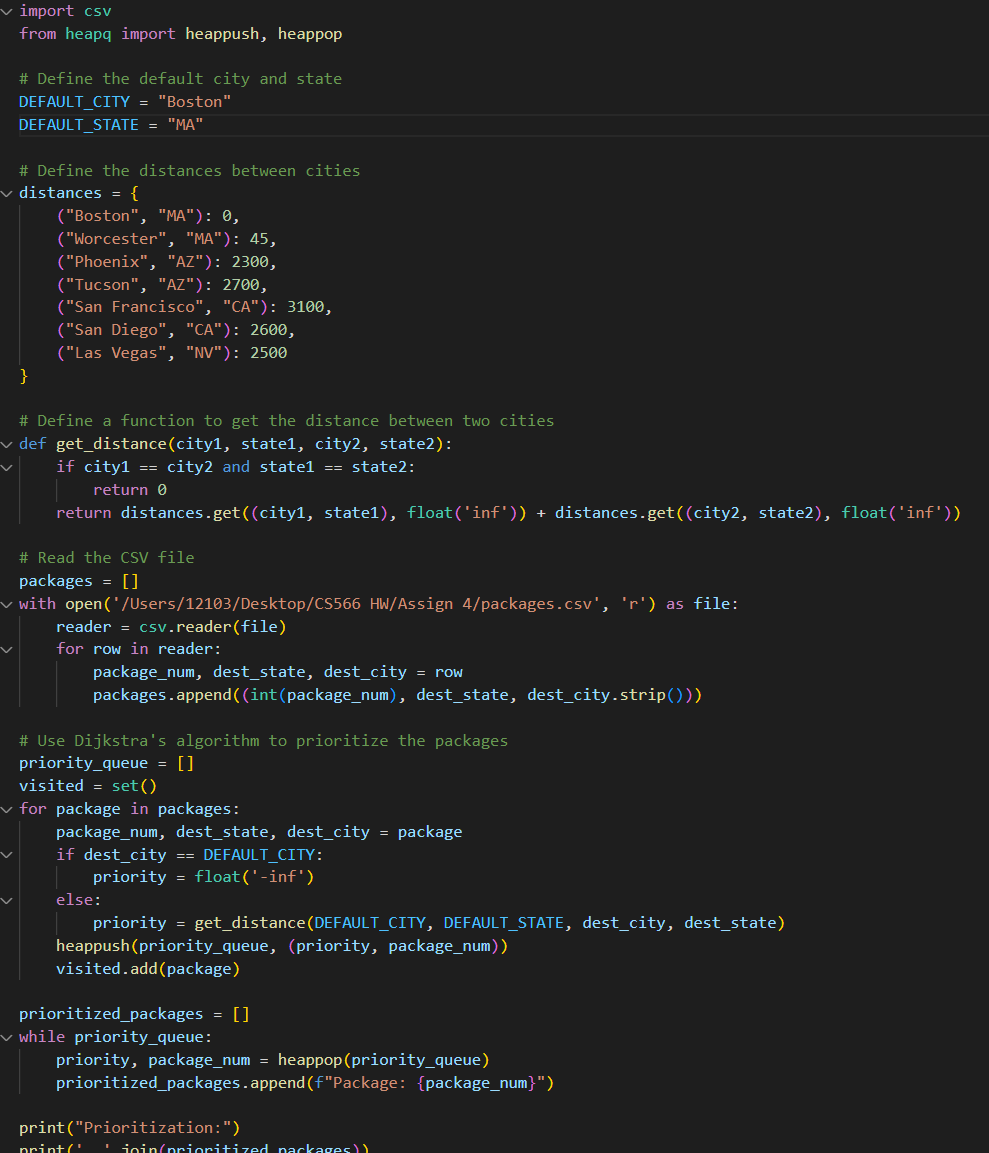
In this assignment, I used the Dijkstra algorithm to find the shortest path, which can guarantee that the package departing from the default city, the estimated distance from the departure city, the higher the priority and the shorter the repeated path.

### 6.3 Time complexity (efficiency)

The time efficiency of Dijkstra's algorithm for this code is O(m log n), where m is the number of edges and n is the number of nodes. In this case, the nodes are cities and the edges are the distances between cities.

In the code, we define the distance between cities as a dictionary, so the time complexity of accessing the distance between two cities is O(1). The priority queue used in the algorithm is implemented using a heap, which has a worst-case time complexity of O(log n) for insertion and removal of elements. The number of elements in the priority queue is at most equal to the number of packages, so the total time complexity of the algorithm is O(m log n), where m is the number of packages and n is the number of cities.

## 7 YOUR CODE



## 8 Evaluation



## References (if used. Each of [1], [2], etc. should occur within the paper above.)

[1]

[2]

## Appendix 1 (if needed; should be referenced above, and will be read as-needed only)

## Appendix 2 (if needed; should be referenced above, and will be read as-needed only)