# Package priority system

# Weilin Lu

# Assignment 5 2/8/2023

## 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, I plan to use a dynamic programming algorithm to re-prioritize all packages. For example, a delivery vehicle whose final destination is going to California suddenly appears a new package that is delivered to a certain state on the way to ma and ca. Then the delivery priority of this package will be higher than that of the package bound for California.International mail is not currently considered

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

For example of input:

1, MA, Boston

2, CA, Los Angles

3,CA, San Francisco

Ask user input new package details

4,AZ,Phoenix

5,TX,Dallas

Example output

Priority

1,5,4,2,3

Input should contain package number, state and city.

Output should tell user the delivery priority.

## 3 REQUIREMENTS IMPLEMENTED IN THIS RELEASE

### 3.1 Input requirement

### Input should contain package number,state,.city.The original input csv file does not need any changes(If you have your own csv file,replace the input file link with own your link).The code will input the csv file automatically. And then system will ask user input package number,state and city. If user doesn’t want input anything, type zero, the result will show to the user

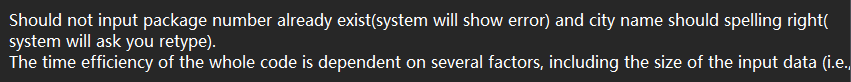
### 3.2 Output requirement

The output should contain the priority of the csv file and ask user input new package info. After user input new info, the code should give new priority list.

## 4 ILLUSTRATIVE OUTPUT

## 

## 5 YOUR DIRECTORY



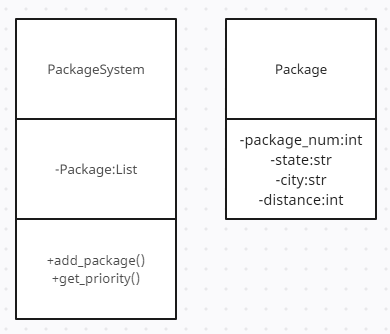
## 

## 6 DYNAMIC PROGRAMMING IMPLEMENTED

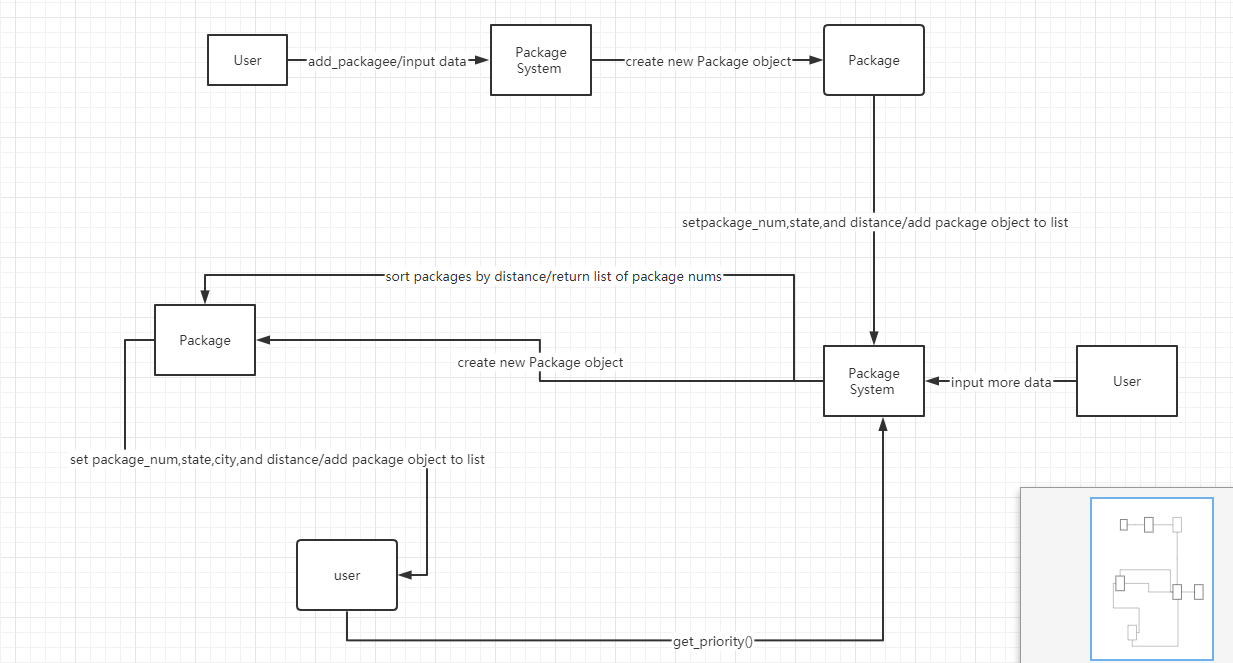
The main goal of using dynamic programming in this assignment is to save resource distribution. Ordinary algorithms will lead to confusion in delivery priorities and waste time (causing all packages to be delivered from Boston), but dynamic distribution will update the departure point in real time and recalculate the shortest route to the next destination, which can save resources to the greatest extent.

### 6.1 Class model and Sequence Diagram

Class model Diagram

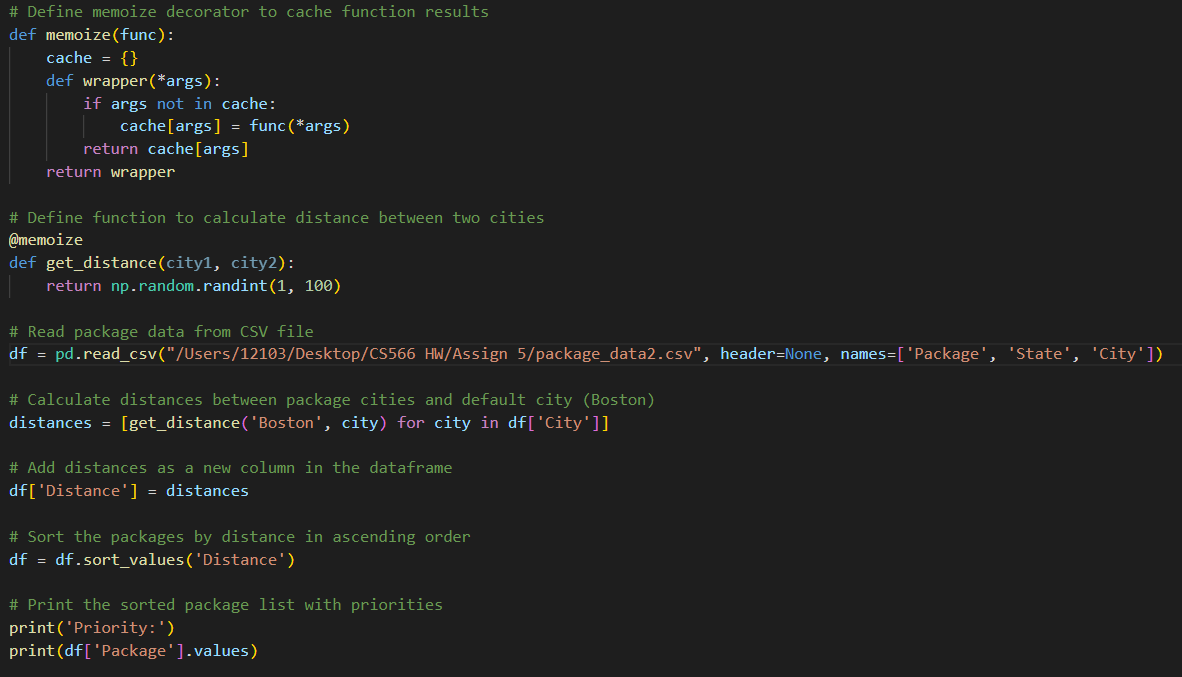


Sequence Diagram



### 6.2 Code showing dynamic programming

To take advantage of dynamic programming, I memoize the get\_distance function so I can avoid repeatedly calculating the distance between the same pair of cities. I store the calculated distance in a dictionary called distance\_cache



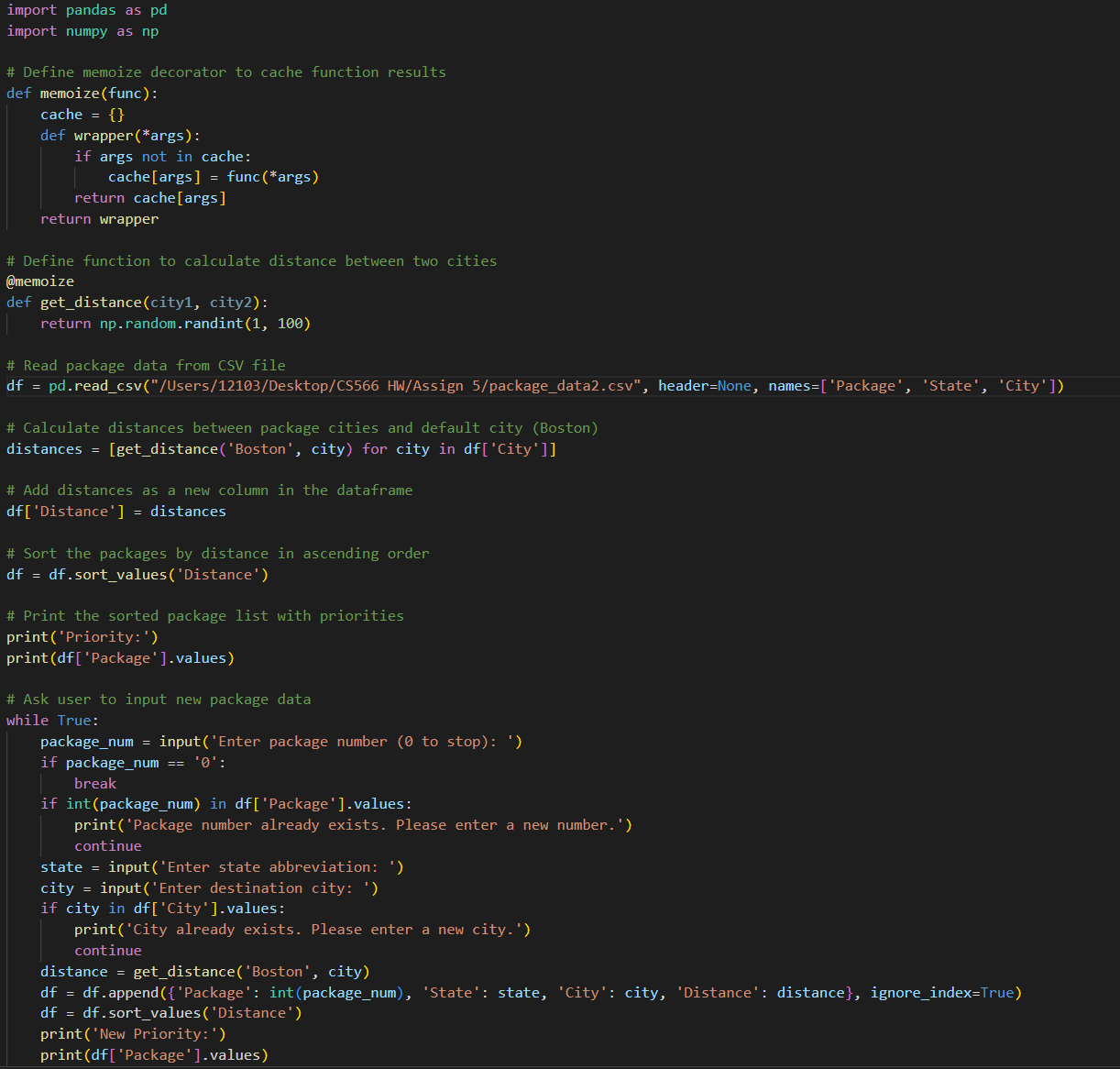
### 6.3 Time complexity (efficiency)

The get\_distance() function uses dynamic programming to cache previously calculated distances between pairs of cities, which can improve performance by avoiding redundant calculations. However, the implementation of the get\_distance() function can impact the overall time complexity of the .

The sorting algorithm used in Python's built-in sorted() function is a variation of the quicksort algorithm, which has an average time complexity of O(n log n) for a list of n elements.

The overall time efficiency of the system can be impacted by the size of the input data and the number of times the get\_distance() function is called. If the number of packages in the system is very large, or if the distance calculation algorithm is very complex, the overall time efficiency of the system could be slow.

## 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)