# PASSENGER CLEARANCE BY SORTING

# Yuhan Xu

## 1 SUMMARY DESCRIPTION

In this part of the homework, according to the passenger destination, I designed the flight release system. According to the obtained passenger's itinerary, the passengers are first divided into domestic and international routes, and according to the domestic or international and the country of the airliner from to do the release. International has the higher prioirty, and if the airliner is from US, it will has the higher priority, other will be placed the order by the first digit of the airliner.Dividing domestic and international, I plan to use the SORT algorithm.

## 2 I/O EXAMPLE FROM PROJECTED COMPLETED PROJECT

Input:(as file)

A,KLAX,LosAngeles,CA,US,AAL

B,KORD,Chicago,IL,US,DAL

C,KJFK,NewYork,NY,US,AAL

D,VHHH,HongKong,China,CPA

E,EGLL,London,London,UK,BAW

F,YSSY,Sydney,Sydney, Australia,AAL

G,PHNL,Honolulu,HI,US,UAL

H,PANC,Anchorage,AK,US,DAL

Output:

National:

Passenger: G

Destination Airport: PHNL

Destination City: Honolulu,HI

Airliner: UAL

Passenger: H

Destination Airport: PANC

Destination City: Anchorage,AK

Airliner: DAL

Passenger: B

Destination Airport: KORD

Destination City: Chicago,IL

Airliner: DAL

Passenger: C

Destination Airport: KJFK

Destination City: NewYork,NY

Airliner: AAL

Passenger: A

Destination Airport: KLAX

Destination City: LosAngeles,CA

Airliner: AAL

International:

Passenger: D

Destination Airport: VHHH

Destination Country: China

Destination City: HongKong,HongKong

Airliner: CPA

Passenger: E

Destination Airport: EGLL

Destination Country: UK

Destination City: London,London

Airliner: BAW

Passenger: F

Destination Airport: YSSY

Destination Country: Australia

Destination City: Sydney,Sydney

Airliner: AAL

Clearance Order:

D E F G H B C A

## 3 REQUIREMENTS

In order to better run the system normally, the requirements of input and output need to be met at the same time. These requirements can make the program better

3.1 Input Requirement:

The package should include the destination ,detailed city, states, contries,and the Airliners they want to take

3.2 Output Requirement:

Based on the passenger information read, the passengers are divided into domestic and international, and the distance is calculated according to the destination, the output is out according to a certain format, and the passengers are released according to the distance.

## 4 ILLUSTRATIVE OUTPUT

Passenger: G

Destination Airport: PHNL

Destination City: Honolulu,HI

Airliner: UAL

Passenger: H

Destination Airport: PANC

Destination City: Anchorage,AK

Airliner: DAL

Passenger: B

Destination Airport: KORD

Destination City: Chicago,IL

Airliner: DAL

Passenger: C

Destination Airport: KJFK

Destination City: NewYork,NY

Airliner: AAL

Passenger: A

Destination Airport: KLAX

Destination City: LosAngeles,CA

Airliner: AAL

International:

Passenger: D

Destination Airport: VHHH

Destination Country: China

Destination City: HongKong,HongKong

Airliner: CPA

Passenger: E

Destination Airport: EGLL

Destination Country: UK

Destination City: London,London

Airliner: BAW

Passenger: F

Destination Airport: YSSY

Destination Country: Australia

Destination City: Sydney,Sydney

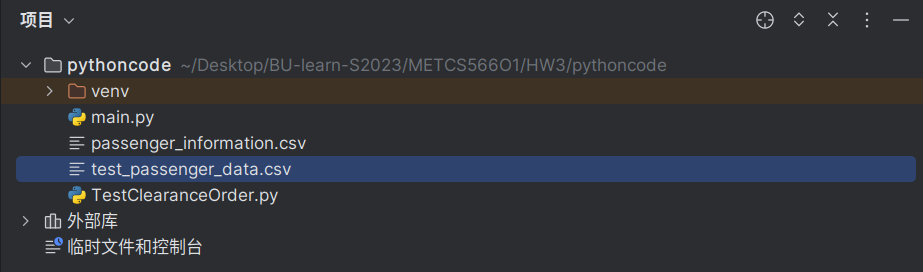
Airliner: AAL

Clearance Order:

D E F G H B C A

The output above is generated by class clearanceOrder, the order is generated by clearanceOrder.sort()

## 5 DIRECTORY



## 6 SPECIALIZED SORTING IMPLEMENTED

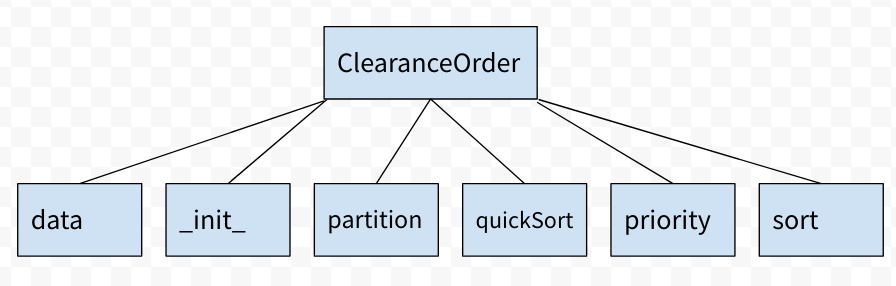
In this project, I used Quicksort to do the distribution of domestic and international passengers, and use Quicksort to release the passengers in order. The time complexity of Quicksort is O (NLOG (N)), which is not very fast.

The reason I use the quicksort are two,

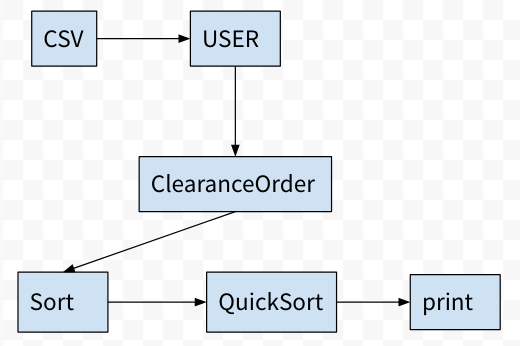
the first reason is that I only know how to write the quicksort,

the second reason is that quicksort is more suitable for my project, because of the separation of national and international.

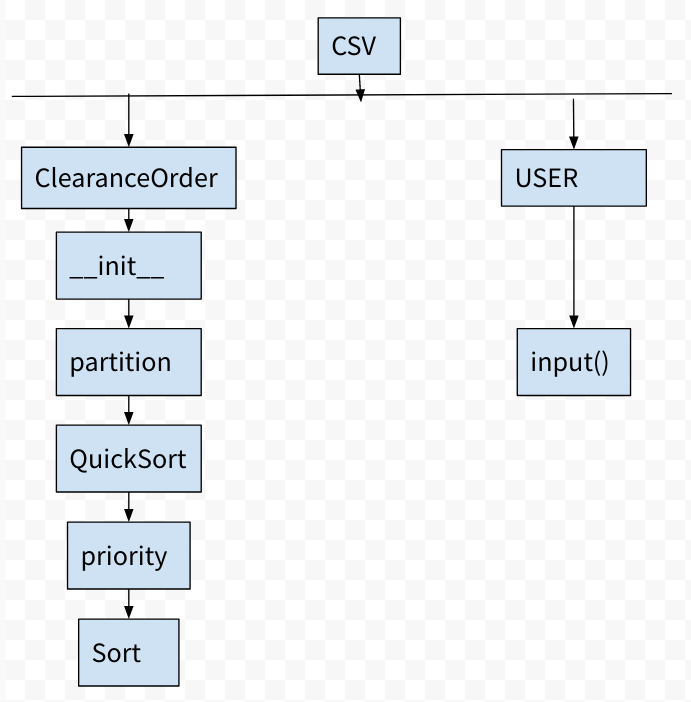
### 6.1 Class model and Sequence Diagram

class model

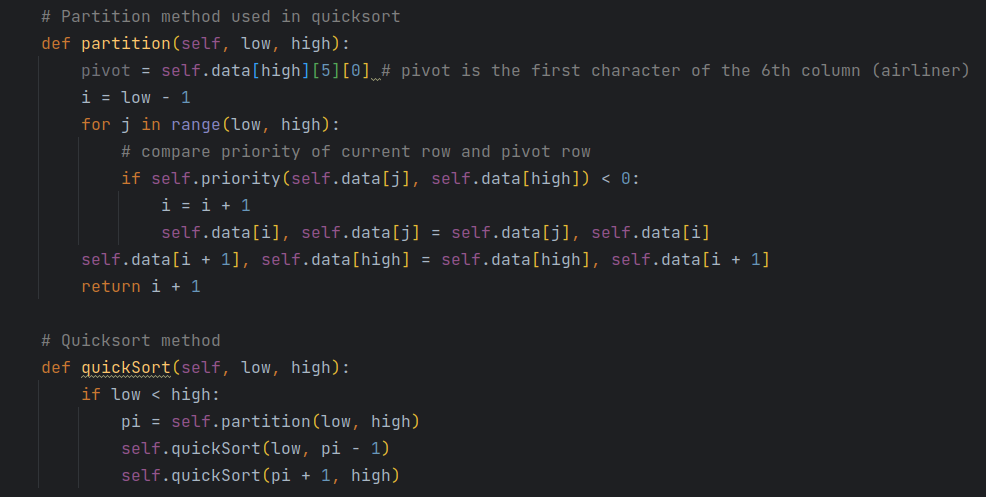
Sequence Diagram

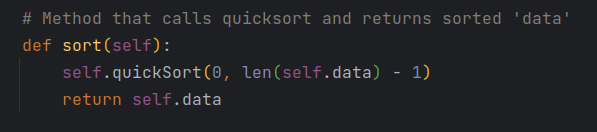


UML



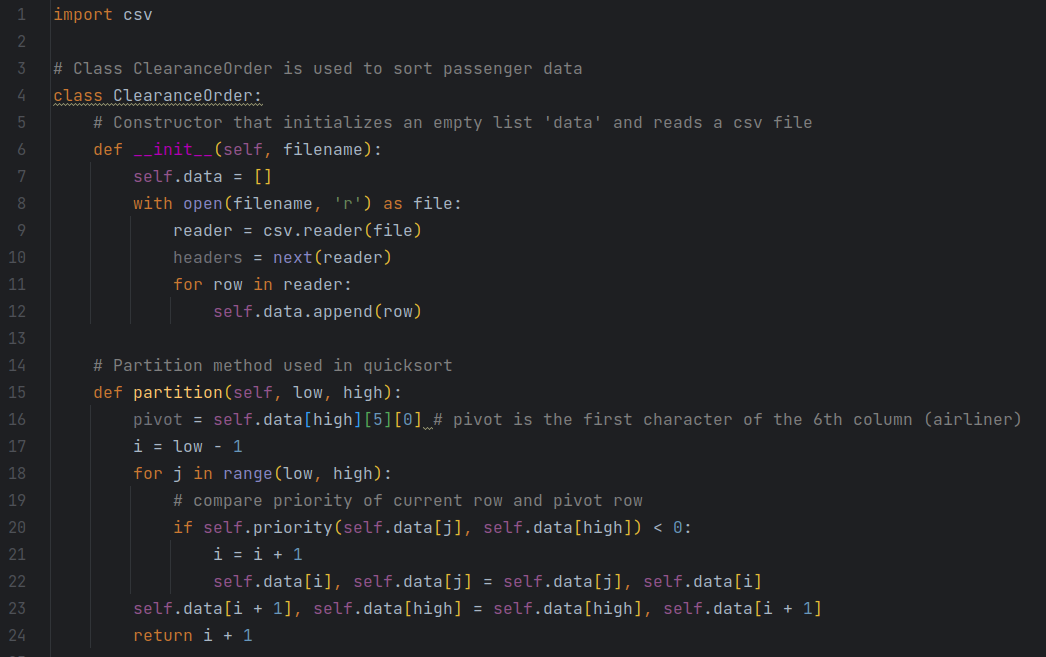
### 6.2 Code showing sorting.

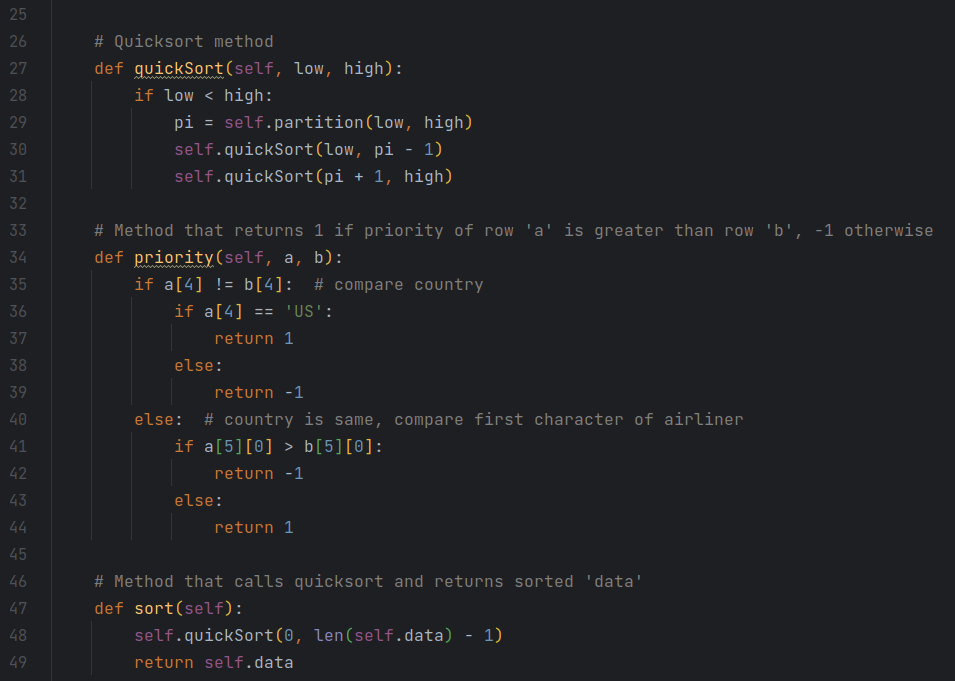


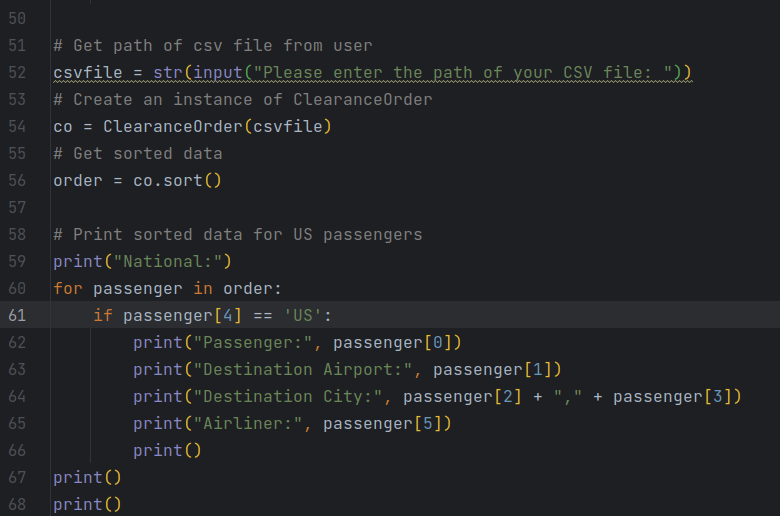


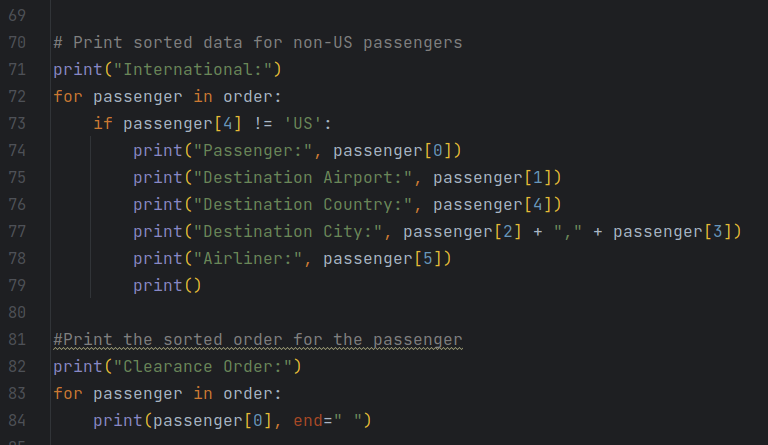
## 7 CODE

main code:









Test case code:

