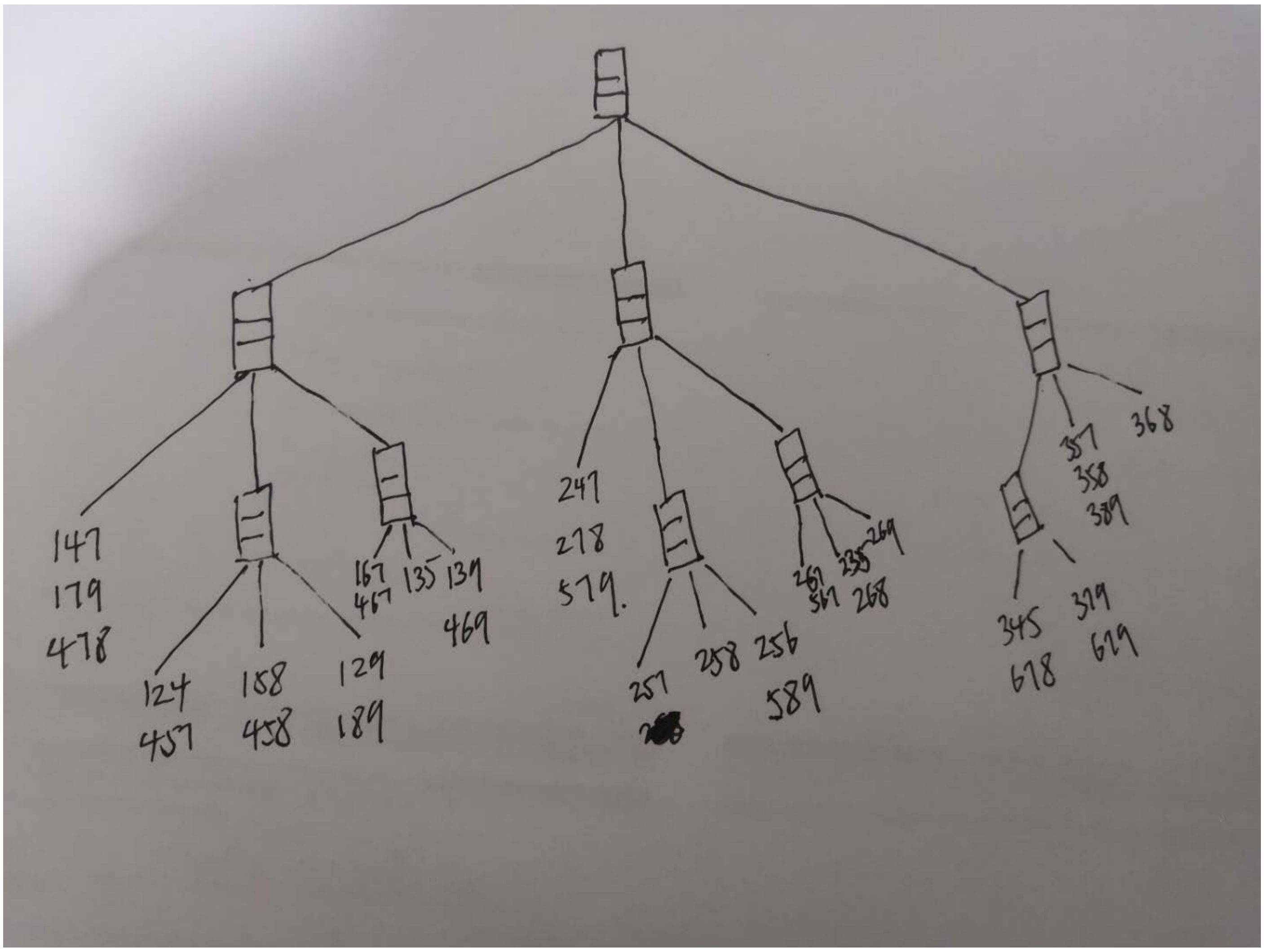
Answer

Q1

(a)

[[[[1, 4, 7], [1, 7, 9], [4, 7, 8]], [[[1, 2, 4], [4, 5, 7]], [[1, 5, 8], [4, 5, 8]], [[1, 2, 9], [1, 8, 9]]], [[[1, 6, 7], [4, 6, 7]], [[1, 3, 5]], [[1, 3, 9], [4, 6, 9]]]]], [[[2, 4, 7], [2, 7, 8], [5, 7, 9]], [[[2, 5, 7]], [[2, 5, 8]], [[2, 5, 6], [5, 8, 9]]], [[[2, 6, 7], [5, 6, 7]], [[2, 3, 5], [2, 6, 8]], [[2, 6, 9]]]]], [[[[3, 4, 7]], [[3, 4, 5], [6, 7, 8]], [[3, 7, 9], [6, 7, 9]]], [[3, 5, 7], [3, 5, 8], [3, 8, 9]], [[3, 6, 8]]]]]

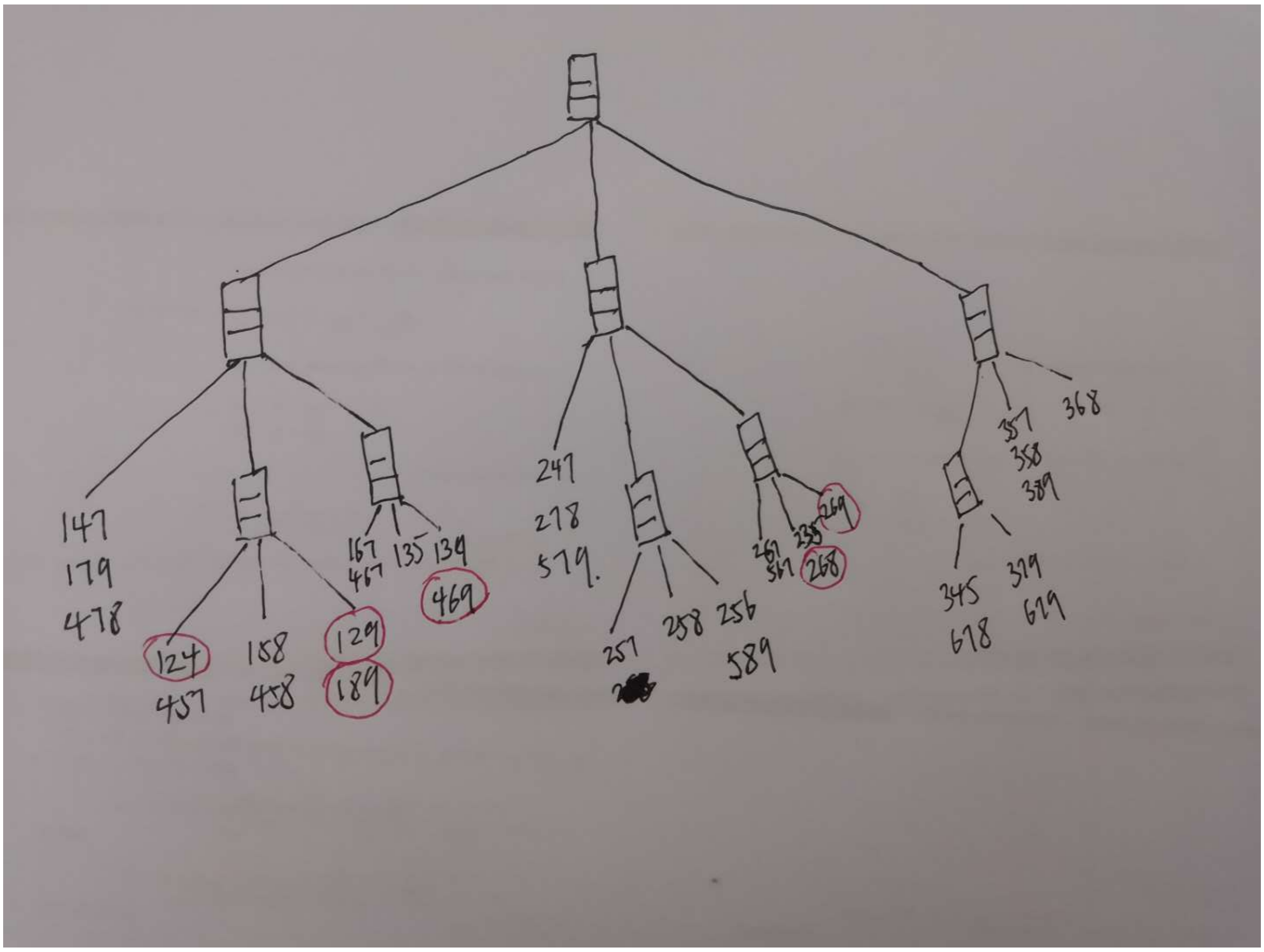
The hash tree is showing as below:



(b)

23 comparisons

And there are 6 candidates showing as below



Q2

(a)

The frequent item set in the csv is showing as below:
And the size of the frequent item set is 61, which means there are 61 frequent item sets with minimize support of 300

{'citrus fruit'}
{'citrus fruit', 'whole milk'}
{'margarine'}
{'coffee'}
{'yogurt'}
{'yogurt', 'whole milk'}
{'rolls/buns', 'yogurt'}
{'yogurt', 'other vegetables'}
{'tropical fruit'}
{'tropical fruit', 'other vegetables'}
{'tropical fruit', 'whole milk'}
{'whole milk'}
{'pip fruit'}
{'cream cheese '}
{'long life bakery product'}
{'other vegetables'}
{'other vegetables', 'whole milk'}
{'butter'}
{'rolls/buns'}
{'rolls/buns', 'other vegetables'}
{'rolls/buns', 'whole milk'}
{'bottled beer'}
{'UHT-milk'}
{'white bread'}
{'bottled water'}
{'bottled water', 'whole milk'}
{'chocolate'}
{'curd'}
{'beef'}
{'soda'}
{'rolls/buns', 'soda'}
{'other vegetables', 'soda'}
{'soda', 'whole milk'}
{'frankfurter'}
{'chicken'}
{'fruit/vegetable juice'}
{'sugar'}
{'newspapers'}
{'pastry'}
{'whole milk', 'pastry'}
{'root vegetables'}
{'other vegetables', 'root vegetables'}
{'whole milk', 'root vegetables'}
{'salty snack'}
{'waffles'}
{'canned beer'}
{'sausage'}
{'sausage', 'rolls/buns'}
{'shopping bags'}
{'brown bread'}
{'napkins'}
{'hygiene articles'}
{'hamburger meat'}
{'berries'}
{'whipped/sour cream'}
{'whipped/sour cream', 'whole milk'}
{'pork'}
{'dessert'}
{'domestic eggs'}
{'frozen vegetables'}
{'onions'}

(b)

```
[['null set	1'], ['other vegetables	185'], ['whole milk	416'], ['other vegetables	168']]
[['null set	1'], ['whole milk	551'], ['other vegetables	219'], ['rolls/buns	59'], ['rolls/buns	94'], ['rolls/buns	131'], ['other vegetables	208'], ['rolls/buns	54']]
[['null set	1'], ['other vegetables	243'], ['whole milk	557'], ['other vegetables	176']]
[['null set	1'], ['rolls/buns	290'], ['other vegetables	54'], ['whole milk	394'], ['other vegetables	94'], ['rolls/buns	87'], ['other vegetables	43'], ['other vegetables	131']]
[['null set	1'], ['other vegetables	238'], ['whole milk	481'], ['other vegetables	228']]
```

Read me

The python environment is python 3.6.
Using the package csv.
The code of Q1 includes the input, the input is the list of dict {} divide by ‘, ’ ,there is a space after the ‘,’ ..
The input of Q2 is the file ‘groceries.csv’ , and should be put in the same folder with the code ‘fp_tree.py’ .
Run code for Q2 can output a csv file called ‘output.csv’ in the code’ s folder and can output the conditional trees larger than 1 in the Terminal at the same time.