**Content**

[Report of Algorithm CA 2](#_Toc142072221)

[Introduction 2](#_Toc142072222)

[Part1 2](#_Toc142072223)

[Question 1 2](#_Toc142072224)

[Question 2 2](#_Toc142072225)

[Question 3 3](#_Toc142072226)

[Question 4 3](#_Toc142072227)

[Part1 4](#_Toc142072228)

[Question 1 4](#_Toc142072229)

[Question 2 4](#_Toc142072230)

# Report of Algorithm CA

### Introduction

This is the report of Algorithm CA report. There are 2 parts, part 1 contains 4 questions, part 2 contains 2 questions. This report contains the designing, timing complexity analysis, running result and source code.

### Part1

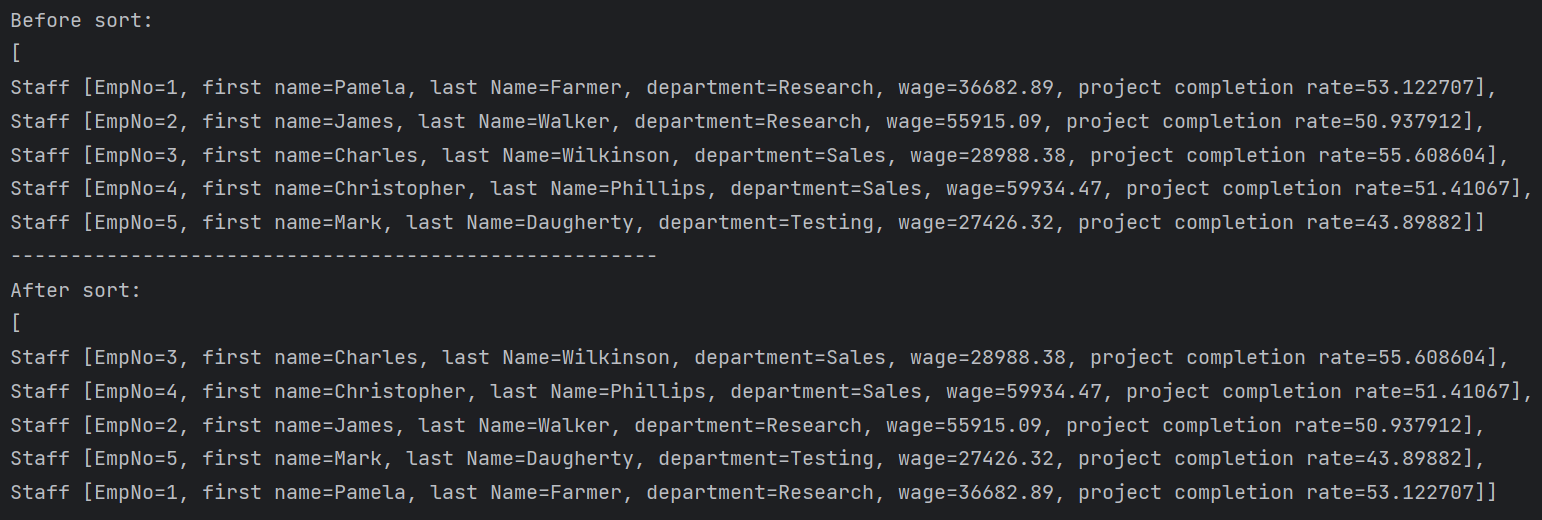
Question 1

**(1) Design:** I implemented the bubble sort method with generic, so that the method can be used not only in sorting of Staff data, but in any other type data.

**(2) Time complexity:** There are 2 for-loops, one inside the other. In the worst case, the calculation times are

So its time complexity is .

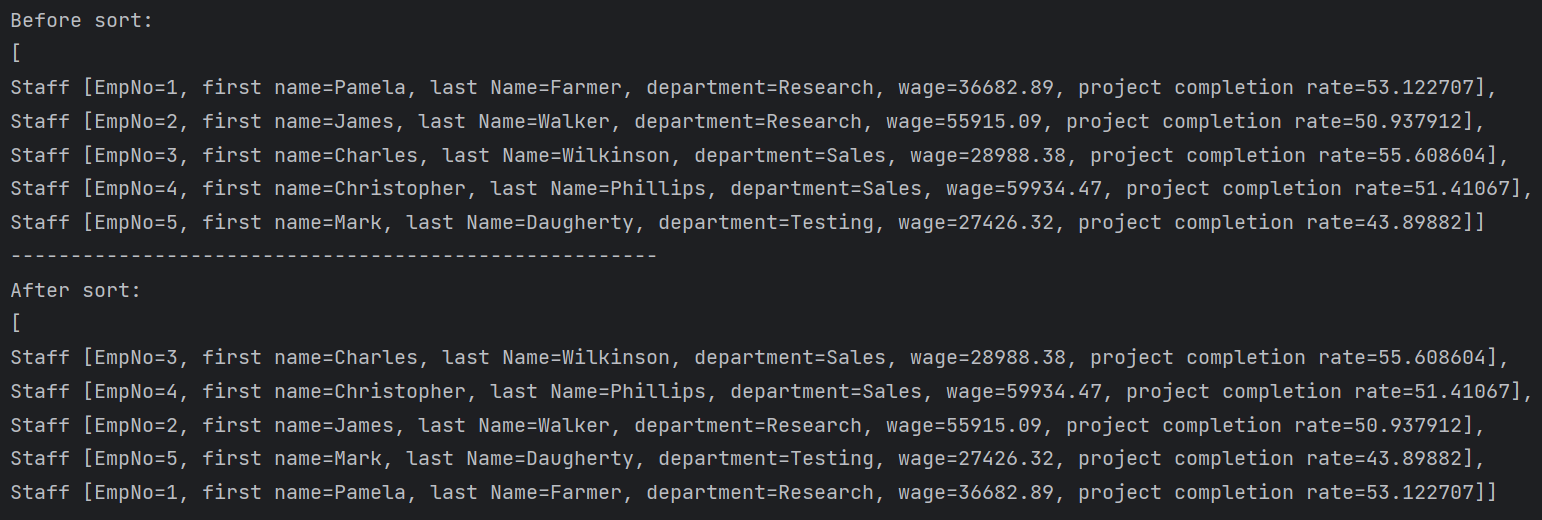
**(3) Code optimization:** I use a Boolean type variable in the outside for-loop to record if the remaining elements are in the correct order. The initial value of the flag is ‘true. Each time the code in the inside for-loop runs, it would change the value of flag into ‘false’. If the value of flag if ‘true’, that means the code in the inside for-loop doesn’t run, that means remaining elements are in the correct order, so break the outside for-loop.

**(4) Result:**

Question 2

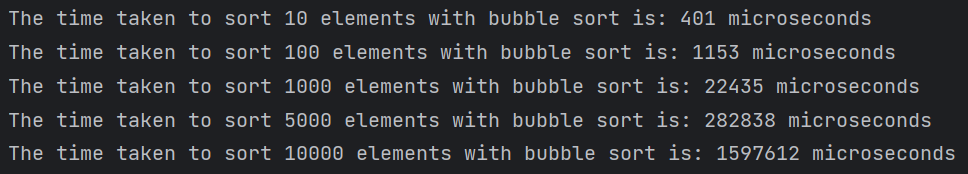
1. **Design:** This quick sort is implemented by 2 methods, one method is the real-quick-sort method, which has 3 parameters (array, begin, end), and the other method invokes this method, and has only one parameter (array). The real-quick-sort method uses recursion.
2. **Time complexity:**  In the worst case, the number of quick sorting operations is n-1 + n-2 + n-3 + ... + 2 + 1, which is the sum of the arithmetic sequence:

The time complexity is . In the average case and the best case, the relevant data in the algorithm analysis shows that the time complexity of quick sorting is .

1. **Result:**

Question 3

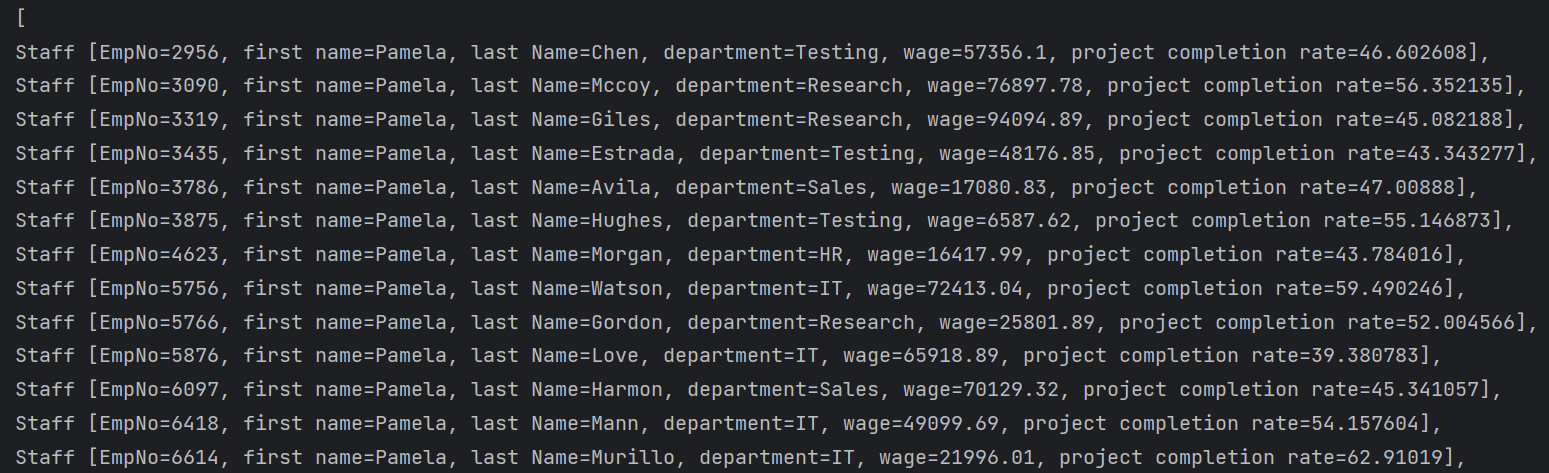
1. **Design:** There are 5 arrays each contains 10, 100, 1000, 5000 and 10000 staffs. Sort them with bubble-sort and record the time cost.
2. **Result:**

****

Question 4

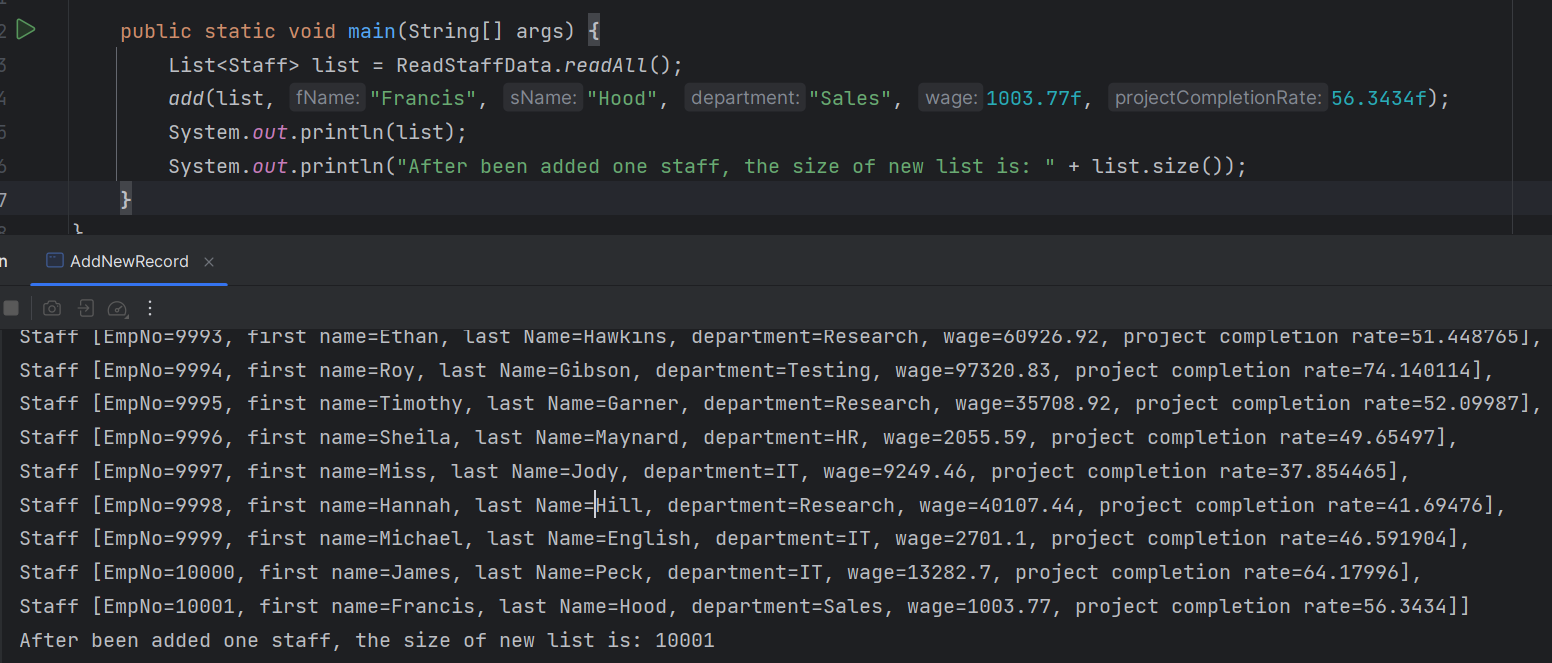
1. **Design:** Template Method Design Pattern is used in the solution of this question. There is an abstract class called ‘AbstractBinarySearch’, there is a ‘find’ method with generic parameter and return. This method is used to search elements from array. There is also an abstract method called ‘compareTo’ with also generic parameters, which is used to compare these 2 parameters. This method is invoked by the ‘find’ method. The implementation class should implement this abstract method.

Template Method Design Pattern is very useful. This abstract class can be used for any this kind of search, each time you just need to implement the compare method.

1. **Result:**

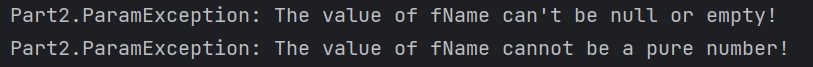
### Part1

Question 1

1. **Design:** Get the count of the array, and use its length as the ID of the new staff.
2. **Result:**

Question 2

1. **Design:** Define a new exception class called ‘ParamException’, which extends the RuntimeException class.
2. **Result:**

****