## Exercise 1：

# Python requires far more than 4GB of RAM to process a 4GB list. This is because lists have many attributes, not just the values of the stored elements. A list will record the value, the data type, the reference number, and the position of the element in the list. Also, it is possible to creating backups of lists during the computing process. In addition, it is also possible that other software is taking up RAM. These are the possible reasons why RAM is running out.

I think the friend can use a computer with larger RAM or use a cloud server. And don't use list, use other data structures like array which only record values to store data.

For calculating the average, the code is modified to reduce RAM consumption by not using weights as variable to store value

文本

描述已自动生成

## Exercise 2：

## Exercise 3：

First I expand the code to a binary tree and test the tree by using the given code.

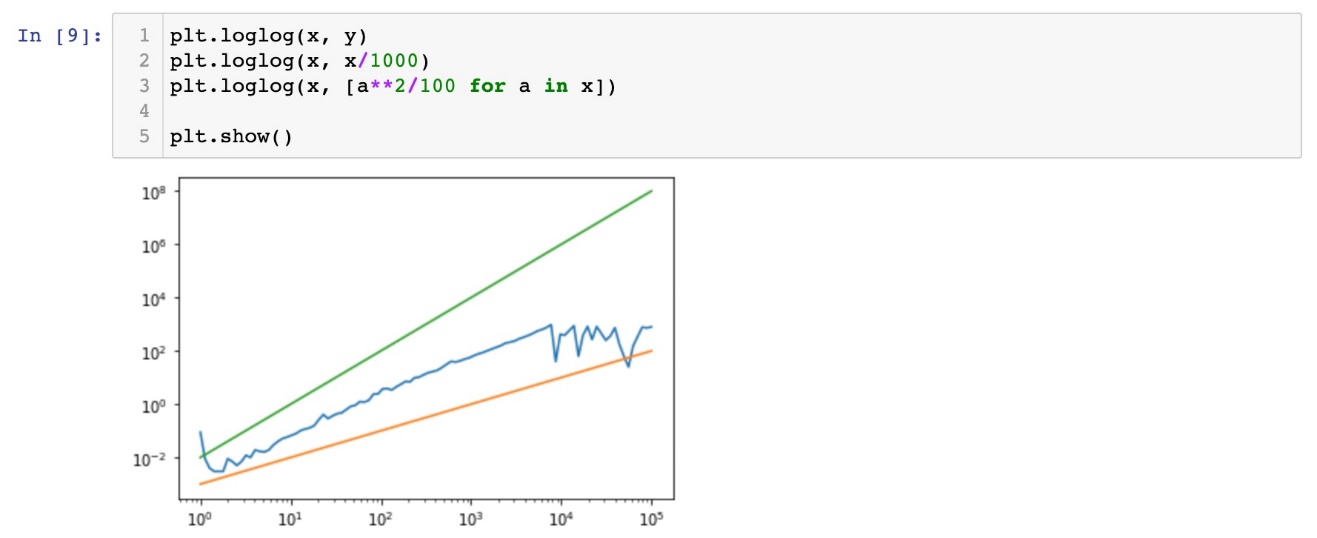
文本

中度可信度描述已自动生成

For sufficiently large in, n starts to flatten图表

低可信度描述已自动生成

The curve of function lies between O(n) and O(n\*\*2), this is the evidence that the time to set up the tree is O(n log n)图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

## Exercise 4：

Both algorithms sort the series in ascending order文本

描述已自动生成

Alg1 will change the position of a number when it is smaller than the previous one until the whole sequence is sorted.

Alg2 splits the entire array until there is only one smallest number in the list. Then two of the lists are compared and the smaller one is merged into the larger one. For the two lists that have been sorted, their minimum values are compared, and then the smaller one and the rest of the numbers are merged into the new list. By recursion, the whole array is sorted.

For data set 1, Algorithm 1 performs better when the data volume is small, but as the data volume becomes larger, Algorithm 2 outperforms Algorithm 1

图表, 折线图

描述已自动生成

For data set 2, Algorithm 1 performs better when data is sorted

图表, 折线图

描述已自动生成

For data set 3, Algorithm 2 performs better

图表, 折线图

描述已自动生成

Only on the ordered dataset, Algorithm 1 performs better. And on any data set, Algorithm 2 has stable performance, so for any other data set, I would prefer Algorithm 2.

Explanation of parallelization of alg2:

To parallelize alg2, we use the map function under multiprocessing. Pool (). After that, the data is divided into two parts and the two parts are sorted at the same time using parallel programming. Finally, we merge the two sorted sub lists together.

Implementation of merge sort in parallel:

# Code:

## Exercise 1:

文本

描述已自动生成

## Exercise 2:

## Exercise 3:

一些文字和图片的手机截图

描述已自动生成文本

描述已自动生成

文本

描述已自动生成图形用户界面, 文本, 应用程序

描述已自动生成

## Exercise 4:文本 中度可信度描述已自动生成 图形用户界面, 文本, 应用程序, 聊天或短信 描述已自动生成文本, 应用程序 描述已自动生成 文本 描述已自动生成 文本 描述已自动生成 文本 描述已自动生成