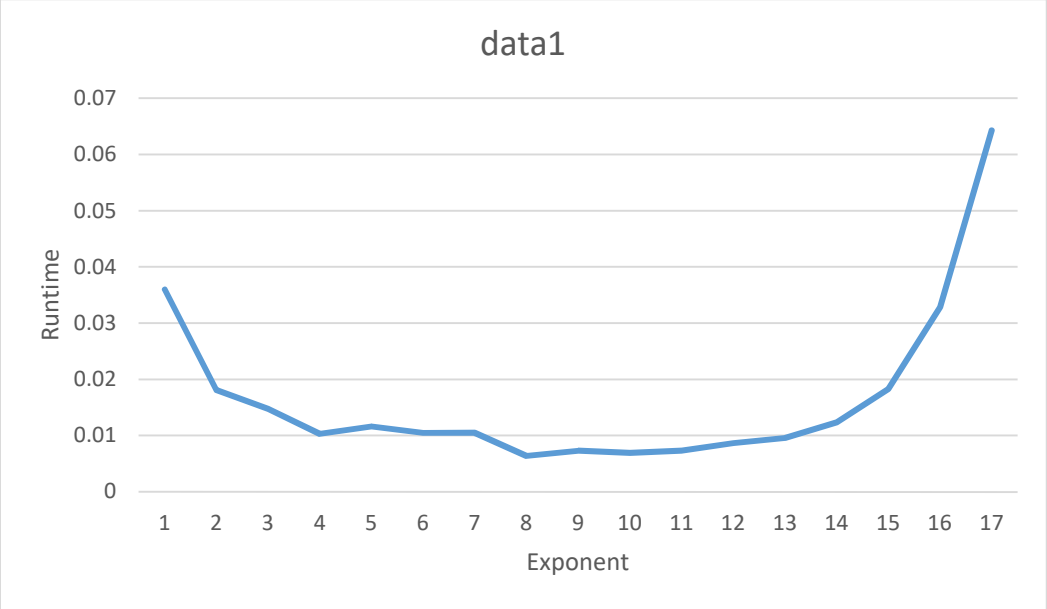
FIT2004 Assignment1

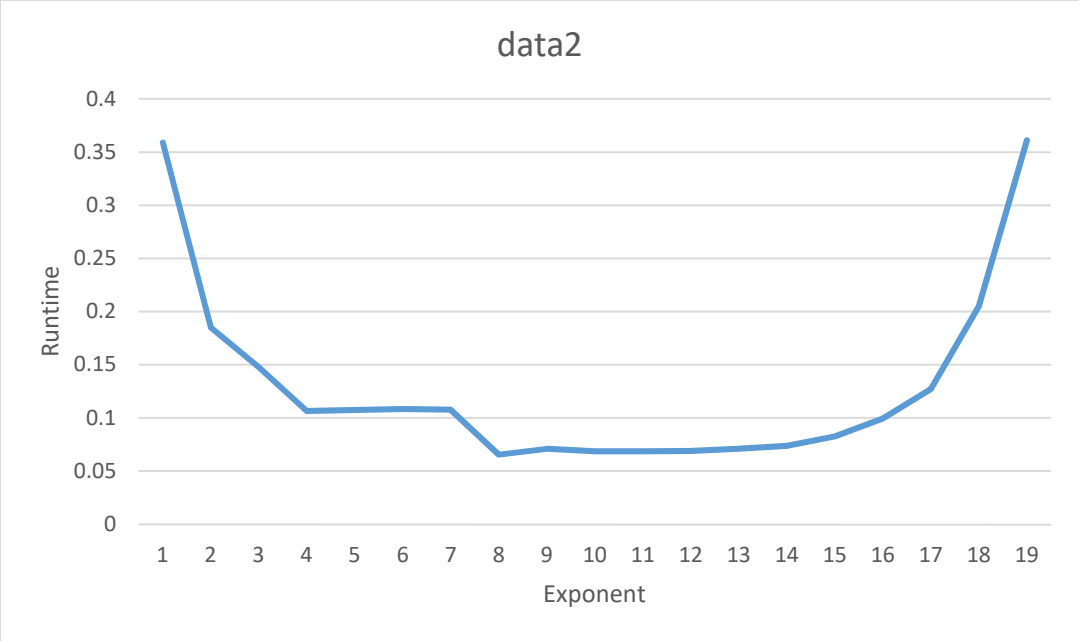
Name: Wong Cheok Foong

StudentID : 29801028

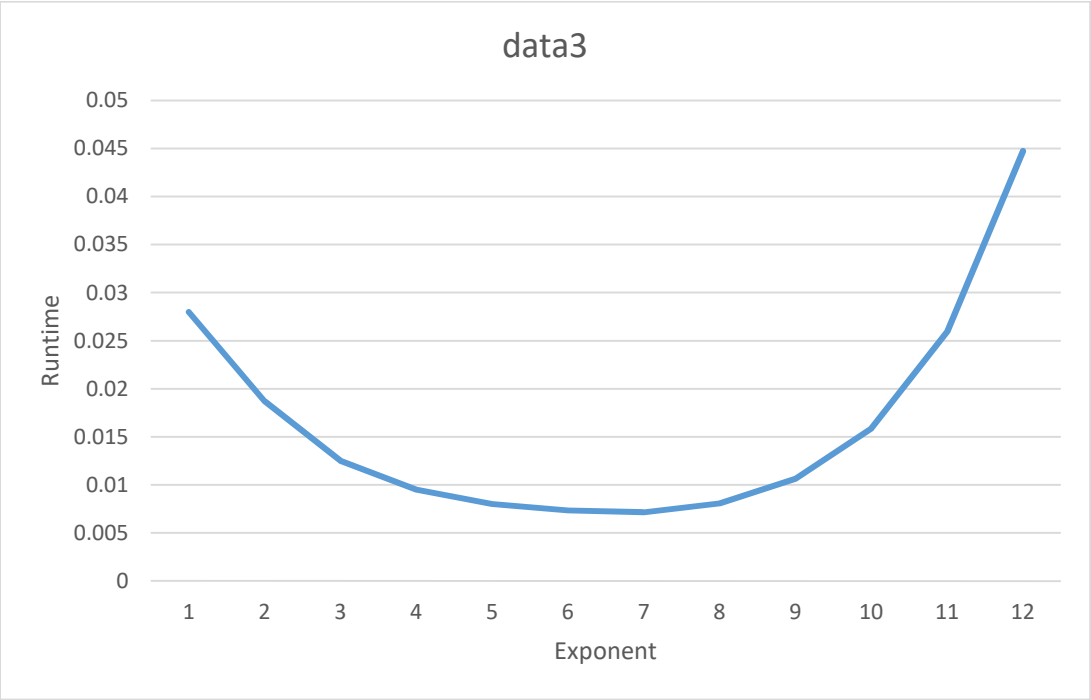
Output for data 1



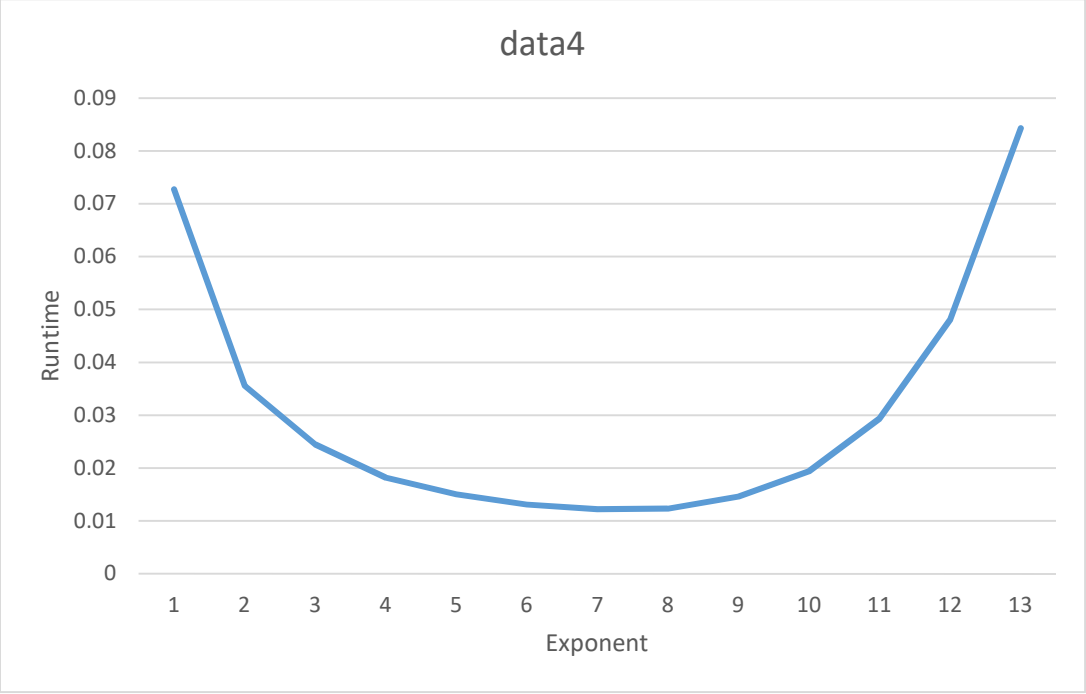
Output for data 2



Output for data3



Output for data4



1. **Identify interesting features of the graphs and explain why they occur**

   All the graphs have a line in the shape of a U and this is because when starting from an exponent of 1 for the power of 2 where 2^1 is 2 which act as the first base, the runtime to sort the list would take a longer time as base 2 only consist of digits 0 and 1 and would require to sort many columns in the form of base 2 as the integers in the input list are very large. As the bases get bigger, the runtime gradually decreases because the number of columns to sort in the respecting bases decreases but only up until a certain point and if the base becomes too large the runtime will increase and this is because the length of the count_array in the radix sort becomes very big corresponding to the number of the base and has to loop through the very large list when trying to find a certain value to append it to the output.

2. **Contrast graphs data1 and data2, and explain the differences/similarities**

   The exponential number where the sorting ends for the graph of data1 would definitely be smaller than the exponential number in graph for data2. Although the range of random integer number to choose from to append into the list is the same but the size of the list is different where the size for the list of data2 is larger by 10 times of that for list of data1. Therefore, data2 would have higher chance to get a random number that is very large and would require an optimal base that is larger than data1. Since the optimal base for data2 is larger therefore the base where the runtime exceeds the initial base would also be larger.

3. **Contrast the graphs from data3 and data4, and explain the differences/similarities**
   The exponential number where the sorting ends for the graph of data 4 is larger than that of data3. Although the range of random integer number to choose from to append into the list is the same but the size of the list is different where the size for the list of data4 is larger than that of data3. Therefore, data4 would have higher chance to get a random number that is very large and would require an optimal base that is larger than data3. Since the optimal base for data4 is larger therefore the base where the runtime exceeds the initial base would also be larger.

4. **Contrast the combined data1 & data2 graphs with the combined data3 & data4 graphs, and explain the differences/similarities**
   The exponential number where the sorting ends for the graph of data 4 and data 3 is much smaller than that of data1 and data2. The difference is that although data 1 and data 2 has much smaller integer numbers in their list compare to data 3 and data 4 but data 1 and data 2 have much larger size of list compare to the other two. We know that in the radix sort we need to create an output list with the same length of input list to be

able to return is O(n) complexity, then we create another count_array list of length of base which is O(b) complexity. When sorting every single column in radix sort will always loop through the count_array list and always create new output list to append the new sorted list and this gives us complexity of O(n+b). The numerical value of the greatest element in num_list is M and has a complexity of O(logM) in radix sort. Since O(logM) is smaller than O(n+b), we will eliminate the smaller complexity which result in O(n +b). Therefore, now comparing back all the four data we know that complexity O(n+b) is prioritize and the runtime for data1 and data 2 is generally higher than the other 2 and the curve of the graph for data 3 and data 4 is also greater.