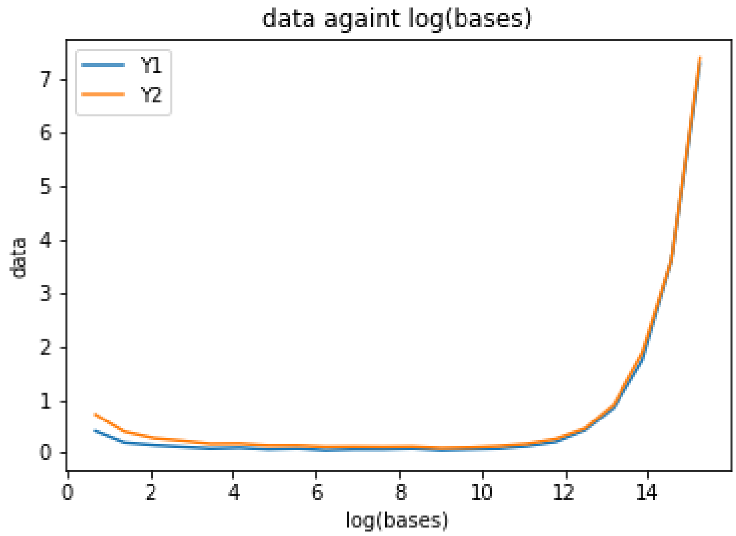
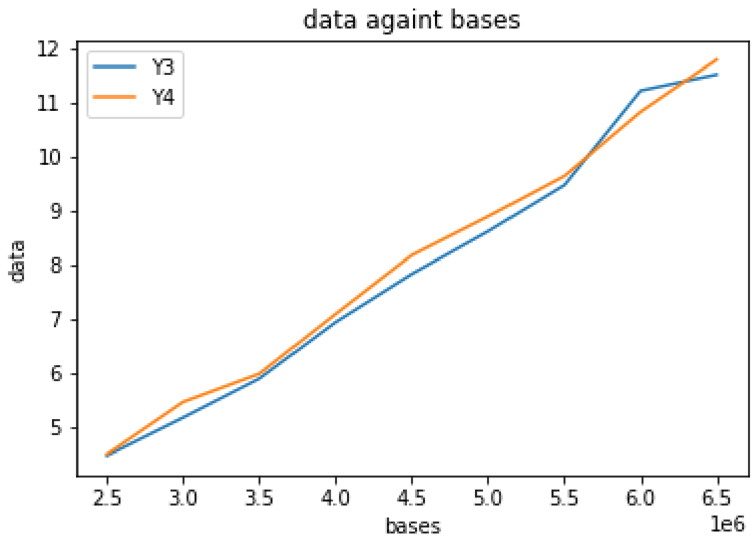
Name: Poon Yeong Shian

ID: 30696003

Graph1: y1 and y2, with log bases1 as the horizontal axis and vertical axis corresponds to the runtimes.



Graph2: y3 and y4, with bases2 as the horizontal axis vertical axis corresponds to the runtimes



Time complexity of radix sort is $O((n + b) \ \log_b M)$ where n is the length of the list, b is the value of base, M is the numerical value of the maximum element of the list.

1. When the base is low, the number of columns produce by $\log_b M$ will be high, so the time taken for the count sort to insert element into count array increases.
   There a region of low time taken in between low base and high base because this region has the optimal base to make the radix sort run faster.
   When the base is high, the number of columns produce by $\log_b M$ will be low, so the time taken for the count sort to insert element into count array decreases.

2. It is because y2 need to compute the 2 times more data than y1 and when base is low, we will have high number of columns. As the result, y2 need twice times to insert the column into count array.

3. It is because when base increases, the number of columns will decreases and which eventually makes the difference of (n + b) * col between y1 and y2 smaller, so they will be less difference in computation time.

4. It is because when base increases, the number of columns will decreases and which eventually makes the difference of (n + b) * col between y3 and y4 smaller, so they will be less difference in computation time.

5. It is because y3 and y4 both using a really large base, so the time to construct the count array is directly proportional the base.