Instruction Set: -

a) Do not use any API of any form to solve the below mentioned problems. Using API's will result in a Zero Credit. The only API acceptable is a graph plotting API.

b) Acceptable programming languages (C, C++, Java, Python, C#, JavaScript)

c) Your code should run and compile. The code that doesn't run won't be graded.

d) Mention in detail how to run your code. For example: -

```
>> python xyz.py <arg1> <arg2> …
>> javac xyz.java
>> java xyz
```

e) Provide the code file and not the compiled file

f) The code should run from terminal/command prompt. No IDE or package should be pre-assumable.

e) Expected files in canvas: -
i) Solution document (word/ any text editor). Handwritten documents won't be accepted

ii) Code file;

iii) Image files (png/jpg) for your plot results;

iv) A tabular representation of *Time vs Size,* from which you got your graph. The table could be attached as an new excel sheet document or could be included in your solution document

PROBLEM SET

1. Implement these three sorting algorithms: -

I) Insertion Sort [*10 points*]

II) Quick Sort with fixed point partitioning (e.g., pivot chosen as first or last index) [*10 points*]

III) Quick Sort with randomized/probabilistic partitioning (pivot chosen at random or based on a model) [*10 points*]

Disclaimer: -

I) Size of list= S <= 10^{10}

II) Numbers could be positive/negative with repetition.

A) Plot the running time of both the algorithm (Time verses Size) between algorithm II and I. Find out the input size for which I overtakes II in terms of time. [*5 points*]

[You could use https://plot.ly/ or your could use an API such as plot.py in python to do the same].

B) Compare algorithm II and III. Plot a graph to compare the results. Which approach seems to be better, if any, and why? [*5 points*]

C) Repeat problem ***A*** with a sorted list as the input and plot your results. [*5 points*]

D) Repeat problem ***A*** with a inversely-sorted list as the input and plot your results. [*5 points*]

2. [*30 points*] IU has invited new grad students for their orientation. In the orientation each student can pick up items from the table, which belongs to a particular category such as T-shirt, Pen, water bottle, etc.

Given the condition that there are infinite categories available and each category has infinite supply of items so that a student can pick up any number of items in a category and as many categories (N) as he/she wants.

Given the above scenario find out the greed-quotient of a particular student.

A student has a Greed Quotient, GQ, if GQ categories have at least GQ number of items and the remaining N - GQ categories have less than GQ items.

For example: -
Number of categories (N)= 5;
Items picked up per category={4,1,7,5,6}.
Answer (GQ)=4.

Explanation: -
The student decides to pick up items from 5 categories and for each category he/she decides to pick 4,1,7,5,6 items respectively. Since the student has 4 categories for which he has at-least 4 items each per category in his/her basket and the remaining N - GQ categories, which is equal to 1, has less than 4 items, his/her greed quotient is 4.

Disclaimer: - There could be multiple possible values of GQ, for which the maximum GQ is taken.

Input format: - <Size of the list> and <list>. For example: -
5
{4,1,7,5,6}

Output format: - <Greed Quotient>. For example:-
4