# ITEC874/COMP733 Report for Assignment 2

*45688516_Adhikari*

## 1 Program execution requirements

### 1.1 Program environment

- **Interface**: Google Collab notebook
- **Platform** : Google Cloud Shared Server
- **Cloud Service:** Python 3 google compute backend engine
- **Programming Language** : Python
- **Packages / Libraries**: Panda , Rtree
- **Ram :** 12 gb
- **Storage :** 50 gb
- **OS :** Ubuntu

### 1.2 Input files and parameters

Two input files **dataset.txt** and **queries.txt** provided for the assignment were used. From those files provided GitHub repository was made and cloned the repository on colab to gain access to the file itself.

The link of GitHub repo is https://github.com/snj-adhikari/r_tree_sample

Used python file handling function open to read the python file.

### 1.3 Other requirements

There is three python package/libraries requirement to run the code.

- **Panda :** Popular open source library with easy to use data structures such as DataFrame( used for this code) , and data analysis tools for Python programming.
- **Time :** This is python module that provides various time-related function. This package has been used to calculate the processing time for query.
- **Rtree :** It is a ctypes python wrapper which provides advanced indexing feature as well as implementation of rtree indexing. This is major package that helps build rtree for dataset and queries result based on that R tree.

## 2. Program documentation

### 2.1 Program organisation

If your assignment involves multiple files and/or classes, please include brief, high level descriptions of each file/class in your program as shown below.

| Class/File Name | Description |
|---|---|
| BigAssData.ipynb | This is python colab notebook / Jupyter notebook file.<br>Contains all the code documentation as well as code cell.<br>It can easily be executed using Jupyter notebook present in anaconda or online platform such as google colab. |

## 2.2 Making DataFrame (Process) Description

- **Creating DataFrame from dataset.txt file**

We were provided with dataset.txt file for our assignment.
Here , the first line of the file provides the no of dataset present. so , we used first line to get total no of dataset and following lines where used to create dataset DataFrame.

Each line of text file was splitted, getting id , x_cord , y_cord from each line.
So index column of DataFrame was replaced using **id** got from the each line and two columns named x_cord , y_cord was made on the DataFrame.

- **Creating DataFrame from queries.txt file**

Here, we have read queris.txt using Open function of python. Then created array based on splitting space on each line of the file. Since, we need co-ordinate x_1 , y_1 , x_2 , y_2 for our rTree package to get point boundaries. The default file provide the structure of x_1 , x_2 , y_1 , y_2 , we had to re-arrange the array converted from splitting the each line to format we needed.

Hence, the resulting DataFrame has column x_1 , y_1 , x_2 , y_2 along with index.

### 2.3 Function Description

| Function Name (parameters) | Description |
|---|---|
| gen_sequence_method (data , queries) | This function takes two parameter **data** and **queries** which is both DataFrame type.<br>This is the manual function, we loop through queries DataFrame and use DataFrame default **loc** function to compare dataset values on dataset DataFrame passed on the function to find the resulting points between the queries co-ordinates. Finally, return the execution time taken by the sequential scan based method to get result on 100 queries.<br>Pandas **DataFrame.loc** attribute access a group of rows and columns by label(s) or a Boolean array in the given DataFrame. On our case each cell is scanned by the function based on our condition and return the cell that satisfies the result. |
| using_rtree_method (data , queries) | This function takes two parameter **data** and **queries** which is both DataFrame type.<br>This is the rTree package based function, we create index from given dataset passed on the function and used passed queries to find the resulting points between the queries co-ordinates. Finally, return the execution time taken by the rTree method to get result on 100 queries. |

On both function **data** and **queries ,** the DataFrame made from file **dataset.txt** and **queries.txt** was passed to compare result and time from both queries.

**Note :** Whole code and code documentation can be found on my original repo – Check it out in https://github.com/snj-adhikari/r_tree_sample