## How to Run the Python Files

This guide will walk you through the steps to run the provided Python scripts for nearest neighbor search algorithms and visualization.

### Prerequisites

Before you begin, ensure you have the following installed:

* **Python 3.x**: You can download it from [python.org](https://www.python.org/downloads/).
* **pip**: Python's package installer (usually comes with Python).

### Library Prerequisites

The R-tree implementation (my\_rtree.py) is a core component. Ensure you have this file in your project directory. The following Python libraries are required and can be installed using pip:

* tqdm: For displaying progress bars during long operations.
* matplotlib: For plotting and visualization.
* numpy: For numerical operations, especially in utils.py for median\_x.

### Setup Instructions

1. Create a Project Directory:  
   It's recommended to create a main directory for your project and place all the provided Python files (task1\_best\_first\_search.py, task1\_divide\_best\_first\_search.py, task1\_main.py, task1\_sequintial\_search.py, utils.py, my\_rtree.py (assuming this is also provided or you have it), and visualisation.py) inside it.
2. Create Data Directories:  
   The scripts expect specific directory structures for datasets and results. Create the following directories within your main project directory:

your\_project\_directory/

├── Task1\_Datasets/

│ ├── parking\_dataset.txt

│ └── query\_points.txt

└── Task1\_Results/

* + Task1\_Datasets/parking\_dataset.txt: This file should contain your main dataset points. Each line should have id x y (e.g., 1 10.5 20.1).
  + Task1\_Datasets/query\_points.txt: This file should contain the query points for which you want to find nearest neighbors. Each line should have id x y.
  + Task1\_Results/: This directory will store the output files generated by the scripts (e.g., all\_results.txt, RTree\_Performance.png).

1. **Example** parking\_dataset.txt **content:**

1 10.0 20.0

2 15.0 25.0

3 5.0 12.0

4 30.0 40.0

1. **Example** query\_points.txt **content:**

101 12.0 22.0

102 8.0 15.0

1. Install Required Libraries:  
   Open your terminal or command prompt, navigate to your project directory, and install the necessary Python packages:

pip install tqdm matplotlib numpy

### Running the Main Search Algorithms

The task1\_main.py script runs all three search algorithms (sequential, best-first, and divide best-first) and saves a summary of their performance.

To run task1\_main.py, use the following command in your terminal from the project directory:

python task1\_main.py --max\_entries 10 --dataset-dir Task1\_Datasets/parking\_dataset.txt --query-dir Task1\_Datasets/query\_points.txt --results-dir Task1\_Results/all\_results.txt

* --max\_entries: (Optional) Specifies the maximum number of entries per R-tree node. Default is 10. You can experiment with different values.
* --dataset-dir: (Optional) Path to your dataset file. Default is Task1\_Datasets/parking\_dataset.txt.
* --query-dir: (Optional) Path to your query points file. Default is Task1\_Datasets/query\_points.txt.
* --results-dir: (Optional) Path to the output file where all results will be saved. Default is Task1\_Results/all\_results.txt.

After execution, a summary file (all\_results.txt by default) will be created in the Task1\_Results directory, containing the performance metrics and nearest neighbor results for each search method.

### Running Individual Search Algorithms (Optional)

You can also run individual search scripts if needed, though task1\_main.py provides a comprehensive overview.

* **Sequential Search:**

python task1\_sequintial\_search.py --dataset-dir Task1\_Datasets/parking\_dataset.txt --query-dir Task1\_Datasets/query\_points.txt --results-dir Task1\_Results/sequential\_search\_results.txt

* **Best-First Search:**

python task1\_best\_first\_search.py --max\_entries 10 --dataset-dir Task1\_Datasets/parking\_dataset.txt --query-dir Task1\_Datasets/query\_points.txt --results-dir Task1\_Results/best\_first\_search\_results.txt

* **Divide Best-First Search:**

python task1\_divide\_best\_first\_search.py --max\_entries 10 --dataset-dir Task1\_Datasets/parking\_dataset.txt --query-dir Task1\_Datasets/query\_points.txt --results-dir Task1\_Results/divide\_best\_first\_search\_results.txt

### Running Visualization

The visualisation.py script helps in understanding R-tree performance with varying max\_entries and visualizing the R-tree structure.

To run visualisation.py, use the following command in your terminal from the project directory:

python visualisation.py --dataset-dir Task1\_Datasets/parking\_dataset.txt

* --dataset-dir: (Optional) Path to your dataset file. Default is Task1\_Datasets/parking\_dataset.txt. This script now expects the dataset to be provided as an argument.

This script will:

1. Read the data points from the specified dataset file.
2. Find the best max\_entries for the R-tree by testing a range of values.
3. Generate a plot (RTree\_Performance.png) showing the total and average time taken for different max\_entries values, and save it to the Task1\_Results directory.
4. Construct an R-tree using the determined best max\_entries.
5. Display a plot of the R-tree's Minimum Bounding Rectangles (MBRs).