

# Python Essentials and Algorithmic Problems

# Python Essentials and Algorithmic Problems

This repository contains solutions and explanations for key algorithmic problems and Python essentials designed for Machine Learning and general programming proficiency. The content is particularly helpful for learners aiming to master Python concepts and algorithmic problem-solving through dynamic programming and recursion.

## Table of Contents

1. Introduction

2. Python Essentials

- Data Types
- Logical Statements and Loops
- Functions
- Exception Handling

3. Algorithmic Problems

- Warming Up Exercises
- Recursion Problems
- Dynamic Programming Challenges

4. Setup and Usage

5. Tasks and Examples

6. Acknowledgements

# Python Essentials and Algorithmic Problems

## ## Python Essentials

### ### 1. Data Types

Python supports various mutable and immutable data types:

- Mutable: list, set, dict
- Immutable: int, float, str, tuple

### ### 2. Logical Statements and Loops

Learn the basics of:

- Logical Statements: if, elif, else
- Loops: for, while, break, continue

### ### 3. Functions

Learn to write reusable and modular code with Python functions:

- Built-in functions like len, sum
- User-defined functions with proper docstrings
- Global and Local Variables

### ### 4. Exception Handling

Master handling errors gracefully using try-except blocks and understand common exceptions like ValueError and ZeroDivisionError.

## ## Algorithmic Problems

# Python Essentials and Algorithmic Problems

## ### Warming Up Exercise

Analyze a dataset of temperatures from Kathmandu airport:

1. Classify temperatures as cold, mild, or comfortable.
2. Convert temperatures from Celsius to Fahrenheit.
3. Group and analyze by time of day.

## ### Recursion Problems

- Sum of Nested Lists: A function to recursively compute the sum of deeply nested lists.
- String Permutations: Generate all unique permutations of a string using recursion.
- Directory Size Calculation: Calculate the total size of a directory structure, including nested subdirectories.

## ### Dynamic Programming Challenges

1. Coin Change Problem: Find the minimum coins needed to make the amount.
2. Longest Common Subsequence: Find the length of the longest subsequence present in two strings.
3. 0/1 Knapsack Problem: Maximize the total value of items without exceeding the weight capacity.

## ## Setup and Usage

1. Clone this repository:

```
git clone https://github.com/yourusername/python-essentials-algorithms.git
```

2. Navigate to the directory:

```
cd python-essentials-algorithms
```

3. Install Python dependencies if needed:

```
pip install -r requirements.txt
```

## Python Essentials and Algorithmic Problems

4. Run examples or solve tasks in the provided tasks.ipynb.

### ## Tasks and Examples

The repository includes:

- Python essentials notebook (python\_essentials.ipynb)
- Algorithms notebook (algorithms.ipynb) covering dynamic programming and recursion challenges.
- Sample data files for tasks like temperature classification.

### ## Acknowledgements

- Siman Giri, Module Leader for "Essentials of Python for Machine Learning," for the structured coursework.
- Inspiration from classic algorithms in computer science for problem-solving tasks.