Universitas Kristen Krida Wacana

Naskah Ujian

UTS ✓ **UAS** □ **Susulan UTS / UAS** **Ganjil/ ~~Genap~~ 2024 / 2025**

Program Studi	**:**	Sistem Informasi	**NIM**	**:**	
Kode - Mata Kuliah	**:**	SIWP1001 - Algoritma dan Pemrograman Dasar			
NID - Nama Dosen	**:**	1933 – Hendrik Tampubolon	**Nama Mahasiswa**	**:**	
Hari / Tanggal	**:**	Kamis			
Waktu Ujian	**:**	13:30 - 16:50 WIB	**Tanda Tangan**	**:**	
Sifat Ujian	**:**	Take home/ Online (Final Project)			
Lembar Jawaban	**:**	Ya / ~~tidak~~			
Kalkulator	**:**	~~Ya~~ / tidak			

Naskah ujian dikumpulkan bersama kertas jawaban ujian !

Perhatian : 1. Taatilah segala peraturan ujian yang telah ditetapkan
2. Pelanggaran terhadap peraturan ujian dikenakan Sanksi Akademik

---

Final Project Guidelines: Comprehensive Study and Implementation of Algorithms

I Introduction

Objectives:
To deepen your understanding of algorithms by exploring their concepts, applications, and implementations. The project is divided into five major categories: Basic Algorithms, Searching Algorithms, Sorting Algorithms, Arrays Algorithms, and Graph Algorithms.

Basic Algorithms:

- Huffman Coding(Compression Algorithm)
- Euclid's Algorithm
- Union Find Algorithm

Sorting Algorithms:

- Insertion Sort
- Selection Sort
- Heap Sort
- Merge Sort
- Quick Sort
- Counting Sort

Arrays Algorithms:

- Kadane's Algorithm
- Floyd's Algorithm (Detection Algorithm)
- KMP Algorithm
- Quick Select Algorithm
- Boyer-More Majority (Vote Algorithm)

Graphs Algorithms

- Kruskal's Algorithm
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Topological Sort Algorithm
- Flood Fill Algorithm
- Lee Algorithm

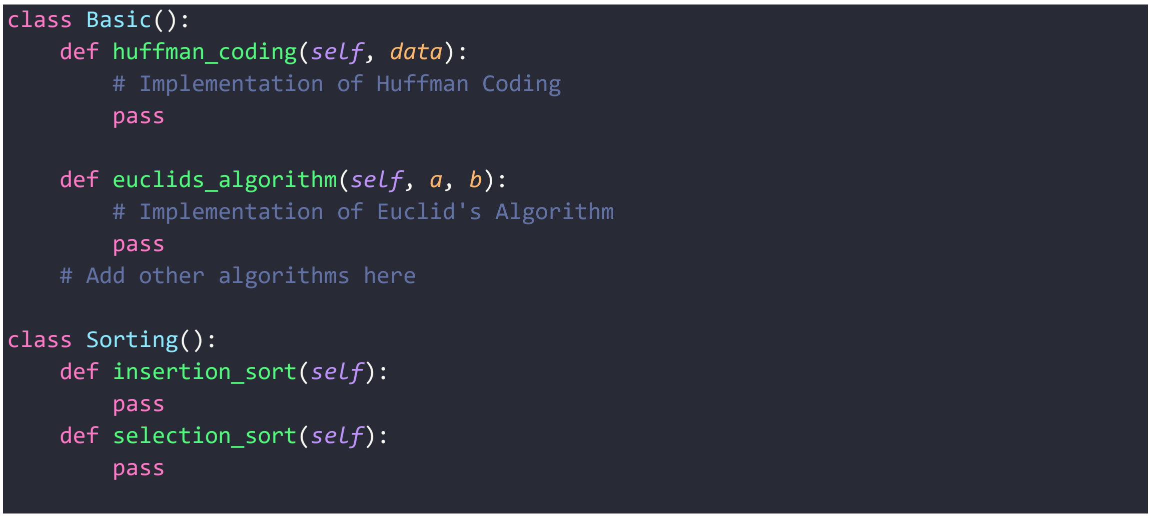
II Tasks

Now your task to do is these following points where each given algorithm

1. Briefly describe the problem statement and its application. Also, explain how the algorithms works (logic and steps)
2. Make a flowchart to visualize the step-by-step process for each algorithm
3. Give the complexity analysis for each algorithm in Big-O notation. You could simply provide a table as the following

Algorithms	Time Complexity			Space complexity
	Best	Average	Worst	
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)

4. Python Implementation. Please do write the algorithm implementation as a Python class for each category, then save it into a file with the name "category.py". For example, basic.py is a python file to do "Basic" category and method for each algorithm. Here is the format and template:



```python
class Basic():
    def huffman_coding(self, data):
        # Implementation of Huffman Coding
        pass

    def euclids_algorithm(self, a, b):
        # Implementation of Euclid's Algorithm
        pass
    # Add other algorithms here

class Sorting():
    def insertion_sort(self):
        pass
    def selection_sort(self):
        pass
```

For more information, please follow the template which already provided in this repo *https://github.com/hendrikTpl/2024-SIWP1001-Final-Project*

III Evaluation

- Code Quality (20%): Clarity, comments, and organization.
- Documentation (20%): Completeness and accuracy of explanations.
- Correctness (20%): Accuracy of algorithm implementation and results.
- Presentation (20%): Clarity and conciseness of the explanation.
- Complexity Analysis (20%): Correctness of time and space complexity.

IV Submition

The submission should include a documentation, code and github repo. Then, you need to compress the file into one single file (.zip .rar, tar.gz) with the name **“Group_{number}_2024_UAS_SIPW1001.zip”**. Please note that, the documentaion can be a markdown, docx, html, or pdf.