



Final Project

Members :

Jason Christopher Chandra

Vincent Joyan

Jose Vallenchee



Final Project

(Presentation's Points)

- Introduction
- Background
- Algorithm Used
- Conclusion
- Reference



Introduction



Introduction

We created 2 python text file compressors with 2 different algorithm, LZ77 and Lempel-Ziv Welch. Then, we will compare these 2 different text file compressor in terms of their runtime speed, space complexity, efficiency and compression ratio.

Problems

Problems

- Nowadays, Everyone saves most of their documents in a digital way. This is good because it means that they don't have to carry physical documents with them and this means that they can view their documents anywhere in the world. But, as we all know digital files will take up a lot of memory space.
- For example, in the 2017 Panama Papers leak, 13.4 million secret documents were leaked to the public. These documents took up to 2 terabytes of data therefore we believe that compressing these file could have saved up a lot of memory space.
- So, For our analysis of algorithms final project, we decided to compare the results between 2 lossless compression algorithms which can be used to compress text files.



Solution



Solution

To create a text file compressor that can greatly reduce the amount of space used



Algorithm Used



Algorithm Used

We will be using 2 algorithms :

1. **LZ77 Encoding:**

Replaces repeated occurrences of data with references to a single copy of that data existing earlier. A match is encoded by a pair of numbers called a *length-distance pair*, which is equivalent to the statement "each of the next *length* characters is equal to the characters exactly *distance* characters behind it in the uncompressed stream".

2. **Lempel-Ziv-Welch Encoding:**

Create a dictionary (a table) of strings used during the communication session. The idea is to parse the sequence into distinct phrases, because when a node is inserted, the code for the current piece becomes the parent node combined with the new character.

Conclusion

Conclusion

As expected, the Lempel-Ziv-Welch algorithm is better at both compression and decompression in terms of both time complexity and space complexity in almost every situation. So, it is more recommended to use the Lempel-Ziv-Welch algorithm compared to the LZ-77.