

PHONE DIRECTOR USING TRI

BY AASHIR AHMED AND WASIQ SHEIKH

WHAT EVEN IS A TRIE?

A Trie data structure is a prefix tree that is used for efficient retrieval of strings. The data is stored as nested dictionaries. It allows for quick searching as its Big-O notation is $O(m)$ where 'm' denotes the length of the string being inserted, searched, or deleted. Due to it being a prefix tree, it allows us to do things like autocomplete which is highly useful in tasks that involve searching strings like perhaps in a search engine.



WHY WE USED IT?

We aimed to create a phone directory that can store multiple contacts and so the functionality of a Trie completely aligned with our implementation needs. A Trie does precisely what we needed to do which was partial searching, adding, storing and deleting contacts. As our phone directory was based on strings, a Trie was the best option in terms of selecting a data structure to use.



OTHER ADTS USED

We used multiple different ADTs such as classes, dictionaries and lists.

The reason for using classes was that it allowed for better readability of the code and allowed us to write coherent and well-linked code.

Dictionaries were used as the data in the Trie was stored as nested dictionaries.

Lists allowed us to ensure our functionality was versatile and covered different aspects of the implementation.



WORK VISION

AASHIR (50%) , WASIQ (50%)

AASHIR (CREATED MOST OF THE BACK-END INCLUDING
THE MAIN TRIE CLASS, DIRECTORY CLASS, THEIR
FUNCTIONS AND OUTPUTS)

WASIQ (CREATED THE FRONT-END AND THE DELETION
FUNCTION FOR THE DIRECTORY INCLUDING
INTEGRATING THE ERROR HANDLING AND OUTPUTS
INTO THE FRONT-END)

THANKS FOR
LISTENING!