Jeffrey Morris

14 Followers

Using π to store data

Jeffrey Morris 1 day ago · 2 min read

Storing data is fairly straight forward. It’s ingested, sometimes compressed, then sent to a database, data warehouse, or data lake.

Data compression algorithms are ways to shrink the size of the data to save on storage costs. They typically look for patterns within the data and optimize by shrinking it.

Another way to potentially save on data storage is to use π as your storage.

Pi is a limitless number representing ratio of a circle’s circumference to its diameter. It also contains every possible number which means it also theoretically stores data for everything within it. Another way to put it is if you can encode it to a numeric, it should exist within Pi.

Encoding any large number into pi would be to store the starting point and ending point of this number within pi, then finding the data between those 2 pointers when decoding.

For example if I wanted to store the number 14159 I would store (1,6) which is far smaller in data than the original 14159.

There’s a few ways we could approach implementing this. The first is to cache the value of pi, then quickly search through it to find your pointers which defeats the purpose of saving storage costs since Pi is being stored. So instead I went with computing the value of Pi on every run which is many orders of magnitude slower.

I also believe that in almost all cases, storing the pointers for larger numbers may actually take more data than the raw data itself.

Python code:
https://github.com/jeffm14/PiEncoder/blob/main/PiEncoder.py
Accepting PR’s, this is the first thing I’ve written in 6 months.

Some quick performance #’s

2 digits — 0 seconds
3 digits — 7.3 seconds

4 digits — 15.1 seconds
5 digits — 9.4seconds
6 digits —Stopped running after 10 minutes

Conclusion: This is a satirical article, this is an awful idea for anyone who wants to store/retrieve data before the world ends. This would be similar to the Monkeysort for data storage. One possible way to expand upon this is to encode data into smaller “Byte” sizes of data chunks.

It would be interesting for encryption purposes since neither of the pointers relate to the actual data that they represent.

Pi Data Database Programming Python