

Python / Coding

1. What is the approach for 3Sum?

Sort the array, fix one element, then use two pointers for the remaining elements. Skip duplicates to return only unique triplets. $O(n^2)$ time.

2. What does sorting mean?

Arranging elements in ascending or descending order. Example:
[-1, 0, -1, 2] $\rightarrow$ [-1, -1, 0, 2].

3. Why do we need sorting in 3Sum?

It allows the two-pointer technique and makes duplicate handling easier.

4. What repetition can we face in 3Sum?

Duplicate values can produce the same triplet multiple times. We skip duplicates for *i*, *left*, and *right*.

5. What is the syntax of `left, right = i + 1, len(nums) - 1`?

Python multiple assignment. It initializes *left* and *right* simultaneously.

6. What are *i*, *left*, and *right* in the first iteration?

If the sorted array has six elements: *i* = 0, *left* = 1, *right* = 5.

7. What is recursion in Python?

A function calling itself to solve a smaller version of the problem. It requires a **base case**.

8. What are decorators?

Functions that modify or extend another function's behavior without changing its source code.

9. How do you create a retry decorator that retries *N* times?

Use a decorator factory that accepts *n*, and a wrapper with a loop and try/except.

10. Why use `@wraps(func)`?

It preserves the original function's metadata, such as its name and docstring. The decorator can work without it.

11. What are `*args` and `**kwargs`?

`*args` accepts variable positional arguments; `**kwargs` accepts variable keyword arguments.

12. Where is the retry attempt limited?

By `range(n)`. For *n*=3, attempt is 0, 1, 2, giving three attempts.

13. What is GIL?

The Global Interpreter Lock in traditional CPython allows only one thread to execute Python bytecode at a time. It limits CPU-bound multithreading but doesn't prevent useful I/O-bound threading.

14. Multithreading vs multiprocessing?

Multithreading uses threads within one process and is generally good for I/O-bound work. Multiprocessing uses separate processes and provides true CPU parallelism.

15. What is a generator?

A function that uses yield to produce values lazily, one at a time, making it memory efficient.

RAG / GenAI

16. Why do we need RAG?

RAG provides the LLM with private, current, domain-specific, and verifiable information without retraining the model.

17. Why RAG if we train and host our own private LLM?

Fine-tuning teaches behavior/domain patterns, but RAG provides reliable access to changing enterprise knowledge and enables source attribution and access control.

18. What are the steps of a basic RAG pipeline?

Documents → Parse → Chunk → Embed → Vector DB → Query → Retrieve → Rerank → Prompt → LLM → Grounded Answer.

19. Can RAG work with an Excel file containing one million rows?

Yes, but don't blindly embed every row. Structured data should generally go into a database and be queried with SQL. Semantic text can additionally use vector search.

20. How would you import one million Excel rows into RAG?

Read in batches, clean/validate, identify structured vs text data, store structured data in a database, embed relevant text, and store embeddings with metadata.

21. How do you chunk an Excel file?

Don't use a RecursiveCharacterTextSplitter blindly. For structured Excel, chunk logically by rows or groups of related rows. For structured numerical data, prefer database/SQL retrieval instead of embeddings.

22. How do you handle multiple versions of the same document?

Store metadata such as document_id, version, effective_date, status, and permissions. Use metadata filtering to retrieve the current/relevant version before semantic retrieval.

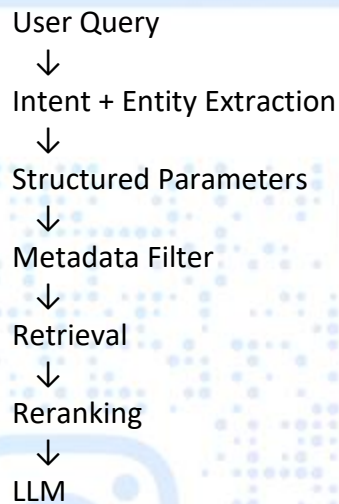
23. How do you retrieve a policy for a specific year, such as 2016?

Extract the year from the query, apply year=2016 as a metadata filter, retrieve relevant chunks, rerank if needed, and send the context to the LLM.

24. How do you implement query intent?

Use an LLM with structured output or a lightweight classifier to extract intent, entities, and filters.

Example:



25. How do you extract intent programmatically?

For simple predictable queries, use rules/keyword matching and regex. For complex queries, use an LLM structured-output classifier.

26. How would you programmatically extract 2015 from “What is the number of leaves in 2015?”

Use regex:

```
import re
```

```
query = "What is the number of leaves in 2015?"  
match = re.search(r'\b(19|20)\d{2}\b', query)
```

```
year = int(match.group()) if match else None
```

27. What is regex?

Regex means **Regular Expression**. It is a pattern used to search, match, or extract text from strings. For example, it can extract years, dates, IDs, or numbers from user queries.