EXPRIMENT 2

s.manoj kumar

(212222040097)

Within a specific use case (e.g., summarizing text, answering technical questions), compare the performance, user experience, and response quality of prompting tools across these different AI platforms.

1.SUMARIZING TEXT

(CHAT GPT) Summary of "The Whispering Banyan"

In a quiet Indian village nestled in the lap of the Western Ghats, life flows gently with the rhythm of nature and tradition. At the heart of this village stands an ancient banyan tree, deeply respected by the locals as a living monument of time. It is not merely a tree, but a silent guardian, believed to have witnessed centuries of human stories—of empires, revolutions, and everyday lives.

The story follows **Arjun**, a young boy full of curiosity and imagination. He spends his days observing the world around him, sketching village scenes—buffaloes in the fields, women drawing water, and temple bells echoing at dusk. One evening during the monsoon season, Arjun seeks shelter beneath the old banyan. As thunder roars and the sky weeps, he begins sketching, unaware that he is on the brink of a life-changing experience.

Amid the wind and rain, Arjun hears a strange sound—not thunder, not rustling leaves, but a **whisper**. It is the banyan tree, speaking directly to him.

"You carry stories," it tells him. "But this land carries more."

Through its whispering voice, the banyan begins to **reveal India's story**—a rich, layered narrative that goes far beyond textbooks. Arjun sees visions of ancient empires like the Mauryas, the grandeur of Mughal courts, the cries for freedom during the independence movement, and modern marvels like space launches. These visions highlight not just historical facts, but the emotional soul of India—its music, struggle, wisdom, and unity in diversity.

The tree doesn't just recount events; it shares India's **cultural spirit**—a blend of tradition, resilience, and transformation. Arjun realizes that the country is not just geography or politics; it is a **living story**, written in many languages, told by many voices.

When the storm subsides, Arjun awakens to find his sketchbook filled with drawings he doesn't remember making—images of what the banyan showed him. Excited, he rushes home to his grandmother, who smiles and tells him, "That tree chose you. India is not just where you live—it's a story you must help tell."

From that moment, Arjun's life changes. He becomes a **storyteller**, traveling from place to place, gathering and sharing the hidden stories of India—from deserts to tea gardens, from temples to city streets. His words bring people together, helping them see not only where they come from, but what they share.

The banyan tree's whisper lives on in Arjun's stories. Through him, the tree's voice reaches others, reminding them that **history is alive**, carried in people's hearts and voices. India, with all its complexity and beauty, becomes more than a nation—it becomes a narrative that never ends.

(CLAUDE) Summary of "The Whispering Banyan"

In "The Whispering Banyan," author Priya Sharma crafts a haunting tale set in colonial India that blends supernatural elements with a poignant exploration of cultural displacement and environmental reverence.

The story centers on Eleanor Hastings, a British botanist who arrives in a remote Indian village in 1887 to catalog local flora for the Royal Botanical Society. Initially viewed with suspicion by villagers, Eleanor is drawn to an ancient banyan tree that dominates the village center. Local lore claims the tree houses ancestral spirits who protect the village, but colonial authorities dismiss such beliefs as superstition.

Eleanor gradually builds a rapport with Deepa, a village elder's daughter who serves as her reluctant guide. Through Deepa, Eleanor learns of the symbiotic relationship between the villagers and the banyan—they protect the tree from harm, and it allegedly wards off disease and drought. Eleanor's scientific skepticism begins to waver when she experiences strange phenomena near the tree: whispers in unknown languages, glimpses of shadowy figures among its aerial roots, and vivid dreams of past centuries.

The narrative tension escalates when Eleanor's superior, Colonel Blackwood, arrives with plans to clear land—including the banyan—for a new administrative compound. Eleanor finds herself caught between her British loyalties and her growing conviction that the tree possesses genuine spiritual significance.

As Eleanor attempts to document the banyan's biological uniqueness to justify its preservation, she experiences increasingly intense visions revealing the tree's centuries-long witness to human history—births, deaths, celebrations, and atrocities. These revelations culminate in a pivotal scene where Eleanor physically interposes herself between colonial soldiers and the tree during an attempted clearing operation.

In the story's climax, a violent storm erupts as axes touch the banyan's bark. Eleanor, standing beneath the tree, experiences a profound mystical connection with it. When lightning strikes nearby, observers report seeing Eleanor seemingly absorbed into the tree itself. The clearing operation is abandoned after several workers report similar supernatural encounters.

The epilogue, set decades later during India's independence movement, shows Deepa as an elder, telling visitors about the tree that now bears unusual markings resembling a woman's face in its bark. Local legend holds that the tree is now guarded by a foreign spirit who joined the ancestral ones, and that it whispers in both local dialects and English.

Sharma's story elegantly weaves historical colonial tensions with themes of ecological reverence and cultural understanding. The banyan serves as both literal setting and powerful metaphor for interconnection across cultures and generations. Through Eleanor's transformation from scientific observer to spiritual participant, the narrative suggests that wisdom comes from embracing rather than explaining mysteries, and that true understanding requires abandoning preconceptions to listen to voices different from our own.

(GEMINI) Summary of "The Whispering Banyan"

"The Whispering Banyan" often features an ancient banyan tree as a central element, imbued with a sense of mystery and the ability to communicate in subtle ways, like whispers. This communication can be interpreted as the tree holding secrets, wisdom, or even the echoes of the past.

In some narratives, characters, often young and curious individuals, are drawn to the banyan tree and attempt to decipher its whispers. These whispers might lead to the uncovering of hidden truths, family secrets, or offer guidance and solutions to problems within a community. The tree can act as a silent witness to history and a repository of local folklore.

The relationship between humans and nature is a recurring theme, with the banyan tree symbolizing the interconnectedness of life and the wisdom that can be found in the natural world. There's often a sense of respect and reverence for the tree, highlighting the importance of understanding and preserving nature's gifts.

Some stories might explore the contrast between skepticism and belief, as characters either try to debunk the myth of the whispering tree or come to embrace its mystical qualities. The whispers can also be a metaphor for intuition, the subtle workings of memory, or the way the past can subtly influence the present.

Ultimately, "The Whispering Banyan" stories often evoke a sense of wonder, mystery, and a deep connection to the natural world, reminding readers of the secrets and stories that might be held within the oldest and most majestic trees around us.

(MISTRAL) Summary of "The Whispering Banyan"

"The Whispering Banyan" is a captivating tale that blends mysticism, tradition, and the power of wisdom. The story revolves around a majestic banyan tree, known as the Whispering Banyan, which stands as a symbol of ancient knowledge and guidance in the heart of various communities.

In the village of North Vania, the Whispering Banyan Tree is a revered entity, its sprawling roots and branches symbolizing the deep-seated wisdom it holds. The tree's whispers carry profound messages that guide the villagers, helping them navigate the complexities of life. EHLA, a young data-accountant warrior, finds himself drawn to the tree's mystical allure. One night, the banyan tree imparts a moral dilemma to EHLA: "Knowledge is power, but wisdom is knowing when to use it." This advice challenges EHLA's belief in transparency and clarity, reminding him that not all truths need to be revealed immediately. As EHLA embraces this wisdom, he begins to see positive changes in the village. Businesses thrive, and relationships improve as people start listening to one another, guided by both the whispers of the banyan tree and the insights of technology

.

In the mystical mangroves of Bhitarkanika, another Whispering Banyan Tree stands as a guardian of forgotten lore and a holder of timeless wisdom. The tree's whispers reveal secrets and test the wisdom of those who seek its counsel. Madhaba, a curious boy, is fascinated by the banyan tree and its ability to unravel mysteries and heal wounds. However, he learns that every secret shared by the tree comes with a price – a fragment of its ancient life force. Madhaba's reverence for the tree grows as he listens to its whispers, which speak of bygone eras, great warriors, love lost, and battles won. The tree's wisdom helps Madhaba solve small problems in the village, earning him respect and admiration from the community

.

In the village of Maya, a traveler seeking stories and wisdom is drawn to the ancient Whispering Banyan Tree. Sitting beneath its sprawling branches, the traveler listens intently to the tree's rustling leaves, which whisper secrets and hidden stories. The traveler shares their own tales of wanderings and discoveries, feeling a deep connection to the village's rich tapestry of narratives. As the sun sets, the whispers of the banyan tree seem to echo the traveler's words, weaving them into the fabric of the village's history

.

"The Whispering Banyan" is a story that emphasizes the importance of wisdom, the power of listening, and the enduring connection between nature and human experience. The banyan tree, with its ancient knowledge and guiding whispers, serves as a reminder that true understanding comes from a balance of tradition and innovation, and that the path to enlightenment often lies in the quiet, contemplative moments beneath its branches.

2.TECHNICAL QUESTIONS

(CHAT GPT)

DATA STRUCTURE

QUESTION:When would you use a hashmap over a binary search tree (BST)?

Use a HashMap when:

> **You need fast lookups, inserts, or deletions** — HashMaps provide **average-case O(1)** time complexity for these operations.
>
> **You don't care about order** — HashMaps are unordered (unless you're using something like LinkedHashMap in Java or OrderedDict in Python).
>
> **You need to associate unique keys with values** — great for things like frequency counts, caching, lookup tables, etc.

Use a BST when:

> You need to maintain elements in a sorted order — BSTs keep keys in sorted order, so you can perform range queries, find min/max, predecessor/successor, etc.
>
> You want ordered traversal (in-order) — which isn't possible with a standard hashmap.
>
> You want to perform operations like floor, ceil, or closest value searches — BSTs are better suited for this.

(CLAUDE)

DATA STRUCTURE

QUESTION:When would you use a hashmap over a binary search tree (BST)?

When to use a hashmap:

When O(1) average-case lookup, insertion, and deletion are critical

When you don't need ordered data traversal

When you have good hash functions for your keys

When memory overhead is acceptable

When you need direct key-value mapping

When to use a BST (especially a balanced one):

When you need ordered traversal or range queries

When you require guaranteed O(log n) worst-case performance

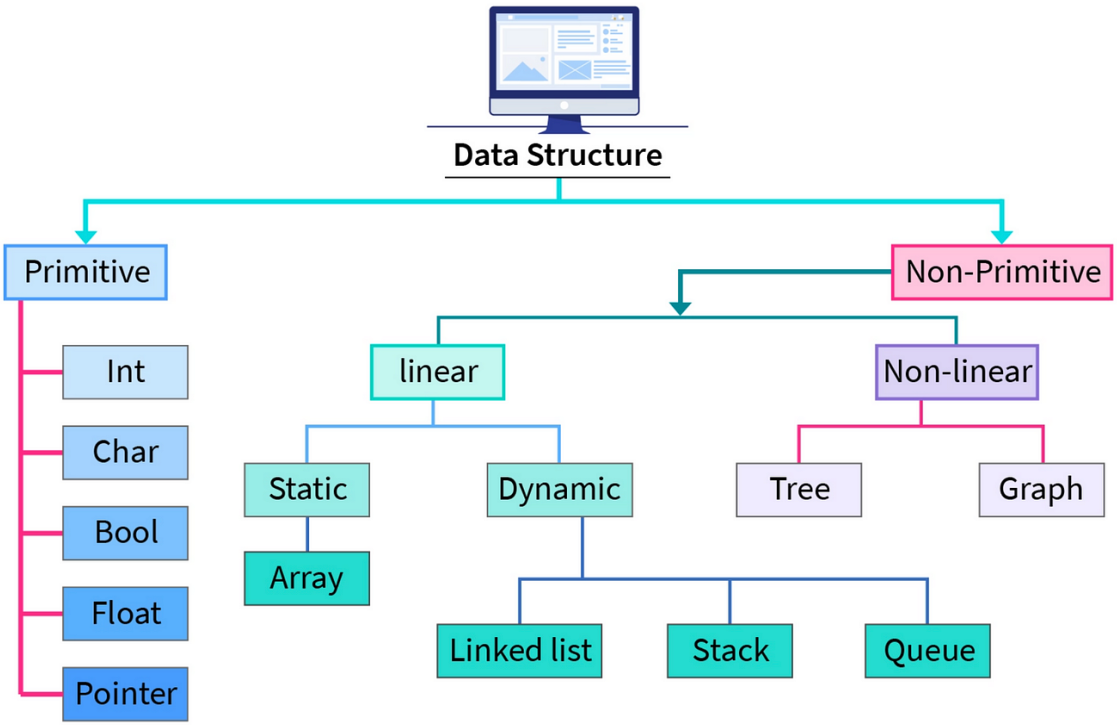
When memory efficiency is a priority

When you don't have good hash functions

When you want to avoid collision handling complexity

(GEMINI)

DATA STRUCTURE



QUESTION:When would you use a hashmap over a binary search tree (BST)?

You would typically use a hashmap over a BST when:

- **Fast Average-Case Lookups, Insertions, and Deletions are Paramount:** Hashmaps offer an average time complexity of O(1) (constant time) for these fundamental operations. This means that regardless of the size of the map, accessing, adding, or removing an element takes roughly the same amount of time on average. BSTs,

on the other hand, have an average time complexity of O(log n) for these operations, which is still efficient but slower than O(1) for large datasets.

- **Order of Elements Doesn't Matter:** Hashmaps do not maintain any specific order of their elements. If the sequence in which you store or retrieve elements is irrelevant to your application, a hashmap is often the better choice due to its speed. BSTs, by their nature, maintain elements in a sorted order (in-order traversal gives sorted keys).
- **You Primarily Need Key-Based Access:** Hashmaps are designed for efficient retrieval based on a key. If your main interaction with the data structure involves looking up values using their unique keys, hashmaps excel. While BSTs also allow key-based access, the average time complexity is higher.
- **Simplicity and Ease of Implementation (in many cases):** For basic key-value storage and retrieval, hashmap implementations are often conceptually simpler and require less code than balanced BST implementations (like AVL trees or red-black trees) which are needed to guarantee O(log n) performance in all cases.

Feature	Hashmap	Binary Search Tree (BST)
Average Lookup	O(1)	O(log n)
Average Insert	O(1)	O(log n)
Average Delete	O(1)	O(log n)
Worst-Case Lookup	O(n) (due to collisions)	O(n) (for skewed trees)
Worst-Case Insert	O(n) (due to resizing/collisions)	O(n) (for skewed trees)
Worst-Case Delete	O(n) (due to collisions)	O(n) (for skewed trees)
Ordering	Not maintained	Elements are sorted (in-order)

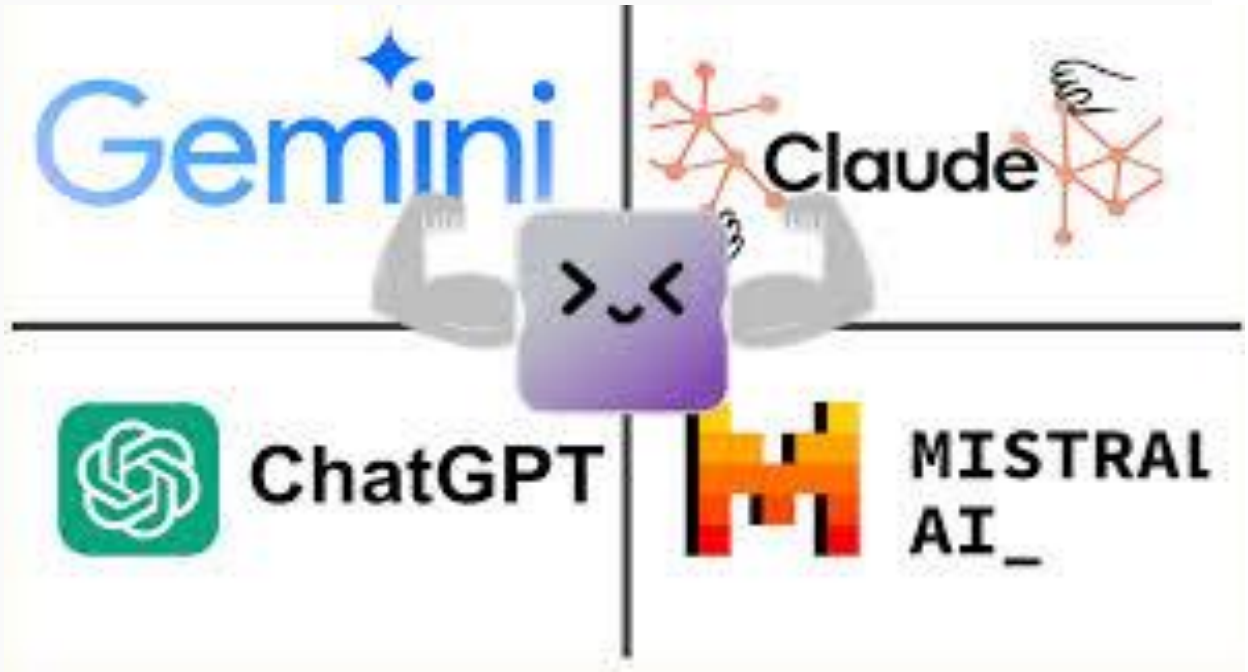
(MISTRAL)

DATA STRUCTURE

QUESTION:When would you use a hashmap over a binary search tree (BST)?

1. **Constant Time Complexity for Lookups**: Hashmaps offer average-case constant time complexity, O(1), for lookups, insertions, and deletions. This makes them very efficient for scenarios where fast access to data is crucial.
2. **Uniform Distribution of Keys**: If the keys are uniformly distributed, a hashmap can provide very efficient performance. However, if there are many collisions, the performance can degrade.
3. **No Need for Ordered Data**: If the order of elements is not important, a hashmap is a good choice. Hashmaps do not maintain any order among the keys, whereas BSTs keep the keys in a sorted order.
4. **Cache Implementation**: Hashmaps are often used to implement caches, where quick lookups, insertions, and deletions are required.
5. **Counting Frequencies**: When counting the frequency of elements (e.g., word count in a document), a hashmap is typically more efficient.
6. **Implementing Associative Arrays**: Hashmaps are ideal for implementing associative arrays or dictionaries, where you need to map unique keys to values.
7. **Space Efficiency**: While hashmaps can be more space-efficient for certain operations, it's important to note that they may require more memory due to the need to handle collisions and maintain a larger array.
8. hashmap might be preferable. BSTs can have O(log n) average case but O(n) worst-case performance if the tree becomes unbalanced.
9. **No Range Queries**: If you do not need to perform range queries (e.g., finding all elements between two values), a hashmap is suitable. BSTs are better for range queries due to their ordered nature.
10. **Simpler Implementation**: For simple use cases, hashmaps can be easier to implement and use compared to BSTs, especially if you use built-in libraries.

3.performance, user experience, and response quality of prompting tools across these different AI platforms.



1. ChatGPT (OpenAI)

Performance:

- **Speed**: Fast, particularly with GPT-4 Turbo.
- **Formatting**: Clean and structured summaries.
- **Reliability**: Very consistent output, handles long context well.

User Experience:

- Intuitive UI; conversation history and memory features enhance experience.
- Temperature & tone not directly adjustable in free plan, but easy to guide with prompts.
- Pro plan allows for GPT-4 Turbo with 128k context—great for long documents.

Response Quality:

- Very strong summarization skills—balances brevity and meaning.
- Captures nuances, especially in academic or technical documents.
- Good at paragraph-level reasoning.

2. Claude

Performance:

- **Speed**: Slightly slower than GPT-4 Turbo, but acceptable.
- **Context**: Extremely strong—can handle up to 200k tokens (ideal for long texts).
- **Formatting**: Simple and readable, but less stylized than GPT outputs.

User Experience:

- Clean interface, minimal distractions.
- Fewer options for fine-tuning style or length in the free UI.
- Prioritizes safe, balanced summaries—often errs on the side of caution.

Response Quality:

- Excellent for faithful summaries with minimal hallucination.
- Tends to preserve original tone and hierarchy of ideas.
- May be overly verbose or cautious unless prompted otherwise.

3. Gemini (Google)

Performance:

- **Speed**: Fast responses.
- **Reliability**: Occasionally cuts off or gives general summaries if prompt isn't specific.
- **Formatting**: Generally clean, sometimes overly concise.

User Experience:

- Integrated with Google Docs and Workspace, good for productivity.
- UI feels more utilitarian than conversational.
- Can sometimes misinterpret long documents without chunking.

Response Quality:

- Gets the gist but sometimes oversimplifies.
- Better for layman summaries than technical fidelity.
- Not as strong in preserving structure of academic documents.

4. Mistral

Performance:

- **Speed**: Fast (smaller models), slower (larger).
- **Reliability**: Depends on host platform—less consistent UI experience.
- **Formatting**: Varies widely—needs more prompt engineering.

User Experience:

- Often open-source interface—less polished than major players.
- Great for developers and tinkerers, not ideal for general users.
- Requires manual work to break down long documents.

Response Quality:

- Adequate for general summaries, but weaker on nuance and structure.
- Often surface-level, especially on complex or technical text.
- Models like Mixtral or Mistral-Instruct show promise but still trail GPT/Claude.

Final Comparison Table

plateform	**performance (Speed/Reliability)**	**UX & Ease of Use**	**Summary Quality**
ChatGPT	⋆ ⋆ ⋆ ⋆ ☆ (fast, polished)	⋆ ⋆ ⋆ ⋆ ⋆ (best-in-class UI)	⋆ ⋆ ⋆ ⋆ ⋆ (balanced, nuanced)
Claude	⋆ ⋆ ⋆ ⋆ ☆ (slower but powerful)	⋆ ⋆ ⋆ ⋆ ☆ (minimalist)	⋆ ⋆ ⋆ ⋆ ☆ (accurate, safe)
Gemini	⋆ ⋆ ⋆ ⋆ ☆ (fast, less context)	⋆ ⋆ ⋆ ⋆ ☆ (well-integrated)	⋆ ⋆ ⋆ ⋆ (broad but light)
Mistral	⋆ ⋆ ⋆ ☆ ☆ (platform dependent)	⋆ ⋆ ☆ ☆ ☆ (developer focus)	⋆ ⋆ ☆ ☆ ☆ (needs refinement)