```
/home/marco/PycharmProjects/Complex/.venv/bin/python
/home/marco/PycharmProjects/Complex/Tests/Case_ToT.py
All modules loaded.
llama_model_loader: loaded meta data with 50 key-value pairs and 201 tensors from ../Models/TinyLlama-
R1-LIMO.Q4_K_S.gguf (version GGUF V3 (latest))
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output.
llama_model_loader: - kv   0:                       general.architecture str              = llama
llama_model_loader: - kv   1:                               general.type str              = model
llama_model_loader: - kv   2:                               general.name str              = Model Limo
R3
llama_model_loader: - kv   3:                         general.size_label str              = 1.1B
llama_model_loader: - kv   4:                            general.license str              = mit
llama_model_loader: - kv   5:                   general.base_model.count u32              = 1
llama_model_loader: - kv   6:                  general.base_model.0.name str              = Tinyllama
STEM Cinder Agent v1
llama_model_loader: - kv   7:               general.base_model.0.version str              = v1
llama_model_loader: - kv   8:          general.base_model.0.organization str              =
Josephgflowers
llama_model_loader: - kv   9:              general.base_model.0.repo_url str              =
https://huggingface.co/Josephgflowers...
llama_model_loader: - kv  10:                      general.dataset.count u32              = 1
llama_model_loader: - kv  11:                     general.dataset.0.name str              = LIMO
llama_model_loader: - kv  12:             general.dataset.0.organization str              = GAIR
llama_model_loader: - kv  13:                 general.dataset.0.repo_url str              =
https://huggingface.co/GAIR/LIMO
llama_model_loader: - kv  14:                          llama.block_count u32              = 22
llama_model_loader: - kv  15:                       llama.context_length u32              = 8192
llama_model_loader: - kv  16:                     llama.embedding_length u32              = 2048
llama_model_loader: - kv  17:                  llama.feed_forward_length u32              = 5632
llama_model_loader: - kv  18:                 llama.attention.head_count u32              = 32
llama_model_loader: - kv  19:              llama.attention.head_count_kv u32              = 4
llama_model_loader: - kv  20:                       llama.rope.freq_base f32              = 10000.000000
llama_model_loader: - kv  21:     llama.attention.layer_norm_rms_epsilon f32              = 0.000010
llama_model_loader: - kv  22:                 llama.attention.key_length u32              = 64
llama_model_loader: - kv  23:               llama.attention.value_length u32              = 64
llama_model_loader: - kv  24:                           llama.vocab_size u32              = 32000
llama_model_loader: - kv  25:                 llama.rope.dimension_count u32              = 64
llama_model_loader: - kv  26:                    llama.rope.scaling.type str              = linear
llama_model_loader: - kv  27:                  llama.rope.scaling.factor f32              = 4.000000
llama_model_loader: - kv  28:                       tokenizer.ggml.model str              = llama
llama_model_loader: - kv  29:                         tokenizer.ggml.pre str              = default
llama_model_loader: - kv  30:                      tokenizer.ggml.tokens arr[str,32000]   = ["<unk>", "
<s>", "</s>", "<0x00>", "<...
llama_model_loader: - kv  31:                      tokenizer.ggml.scores arr[f32,32000]   =
[-1000.000000, -1000.000000, -1000.00...
llama_model_loader: - kv  32:                  tokenizer.ggml.token_type arr[i32,32000]   = [3, 3, 3, 6,
6, 6, 6, 6, 6, 6, 6, 6, ...
llama_model_loader: - kv  33:                tokenizer.ggml.bos_token_id u32              = 1
llama_model_loader: - kv  34:                tokenizer.ggml.eos_token_id u32              = 2
llama_model_loader: - kv  35:            tokenizer.ggml.unknown_token_id u32              = 0
llama_model_loader: - kv  36:            tokenizer.ggml.padding_token_id u32              = 0
llama_model_loader: - kv  37:               tokenizer.ggml.add_bos_token bool             = true
llama_model_loader: - kv  38:               tokenizer.ggml.add_eos_token bool             = false
llama_model_loader: - kv  39:                    tokenizer.chat_template str              = {% for
message in messages %}\n{% if m...
llama_model_loader: - kv  40:            tokenizer.ggml.add_space_prefix bool             = false
llama_model_loader: - kv  41:               general.quantization_version u32              = 2
llama_model_loader: - kv  42:                          general.file_type u32              = 14
llama_model_loader: - kv  43:                                general.url str              =
https://huggingface.co/mradermacher/T...
llama_model_loader: - kv  44:              mradermacher.quantize_version str              = 2
llama_model_loader: - kv  45:                  mradermacher.quantized_by str              = mradermacher
llama_model_loader: - kv  46:                  mradermacher.quantized_at str              = 2025-02-
12T03:41:15+01:00
llama_model_loader: - kv  47:                  mradermacher.quantized_on str              = rich1
llama_model_loader: - kv  48:                         general.source.url str              =
https://huggingface.co/Josephgflowers...
llama_model_loader: - kv  49:                  mradermacher.convert_type str              = hf
llama_model_loader: - type  f32:   45 tensors
llama_model_loader: - type q4_K:  149 tensors
llama_model_loader: - type q5_K:    6 tensors
llama_model_loader: - type q6_K:    1 tensors
```

```
print_info: file format = GGUF V3 (latest)
print_info: file type   = Q4_K - Small
print_info: file size   = 609.53 MiB (4.65 BPW)
init_tokenizer: initializing tokenizer for type 1
load: control token:      0 '<unk>' is not marked as EOG
load: control token:      2 '</s>' is not marked as EOG
load: control token:      1 '<s>' is not marked as EOG
load: special_eos_id is not in special_eog_ids - the tokenizer config may be incorrect
load: special tokens cache size = 3
load: token to piece cache size = 0.1684 MB
print_info: arch             = llama
print_info: vocab_only       = 0
print_info: n_ctx_train      = 8192
print_info: n_embd           = 2048
print_info: n_layer          = 22
print_info: n_head           = 32
print_info: n_head_kv        = 4
print_info: n_rot            = 64
print_info: n_swa            = 0
print_info: n_embd_head_k    = 64
print_info: n_embd_head_v    = 64
print_info: n_gqa            = 8
print_info: n_embd_k_gqa     = 256
print_info: n_embd_v_gqa     = 256
print_info: f_norm_eps       = 0.0e+00
print_info: f_norm_rms_eps   = 1.0e-05
print_info: f_clamp_kqv      = 0.0e+00
print_info: f_max_alibi_bias = 0.0e+00
print_info: f_logit_scale    = 0.0e+00
print_info: n_ff             = 5632
print_info: n_expert         = 0
print_info: n_expert_used    = 0
print_info: causal attn      = 1
print_info: pooling type     = 0
print_info: rope type        = 0
print_info: rope scaling     = linear
print_info: freq_base_train  = 10000.0
print_info: freq_scale_train = 0.25
print_info: n_ctx_orig_yarn  = 8192
print_info: rope_finetuned   = unknown
print_info: ssm_d_conv       = 0
print_info: ssm_d_inner      = 0
print_info: ssm_d_state      = 0
print_info: ssm_dt_rank      = 0
print_info: ssm_dt_b_c_rms   = 0
print_info: model type       = 1B
print_info: model params     = 1.10 B
print_info: general.name     = Model Limo R3
print_info: vocab type       = SPM
print_info: n_vocab          = 32000
print_info: n_merges         = 0
print_info: BOS token        = 1 '<s>'
print_info: EOS token        = 2 '</s>'
print_info: UNK token        = 0 '<unk>'
print_info: PAD token        = 0 '<unk>'
print_info: LF token         = 13 '<0x0A>'
print_info: EOG token        = 2 '</s>'
print_info: max token length = 48
load_tensors: layer   0 assigned to device CPU
load_tensors: layer   1 assigned to device CPU
load_tensors: layer   2 assigned to device CPU
load_tensors: layer   3 assigned to device CPU
load_tensors: layer   4 assigned to device CPU
load_tensors: layer   5 assigned to device CPU
load_tensors: layer   6 assigned to device CPU
load_tensors: layer   7 assigned to device CPU
load_tensors: layer   8 assigned to device CPU
load_tensors: layer   9 assigned to device CPU
load_tensors: layer  10 assigned to device CPU
load_tensors: layer  11 assigned to device CPU
load_tensors: layer  12 assigned to device CPU
load_tensors: layer  13 assigned to device CPU
load_tensors: layer  14 assigned to device CPU
```

```
load_tensors: layer  15 assigned to device CPU
load_tensors: layer  16 assigned to device CPU
load_tensors: layer  17 assigned to device CPU
load_tensors: layer  18 assigned to device CPU
load_tensors: layer  19 assigned to device CPU
load_tensors: layer  20 assigned to device CPU
load_tensors: layer  21 assigned to device CPU
load_tensors: layer  22 assigned to device CPU
load_tensors: tensor 'token_embd.weight' (q4_K) (and 200 others) cannot be used with preferred buffer
type CPU_AARCH64, using CPU instead
load_tensors:   CPU_Mapped model buffer size =   609.53 MiB
llama_init_from_model: n_seq_max     = 1
llama_init_from_model: n_ctx         = 2048
llama_init_from_model: n_ctx_per_seq = 2048
llama_init_from_model: n_batch       = 512
llama_init_from_model: n_ubatch      = 512
llama_init_from_model: flash_attn    = 0
llama_init_from_model: freq_base     = 10000.0
llama_init_from_model: freq_scale    = 0.25
llama_init_from_model: n_ctx_per_seq (2048) < n_ctx_train (8192) -- the full capacity of the model will
not be utilized
llama_kv_cache_init: kv_size = 2048, offload = 1, type_k = 'f16', type_v = 'f16', n_layer = 22,
can_shift = 1
llama_kv_cache_init: layer 0: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 1: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 2: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 3: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 4: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 5: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 6: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 7: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 8: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 9: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 10: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 11: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 12: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 13: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 14: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 15: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 16: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 17: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 18: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 19: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 20: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init: layer 21: n_embd_k_gqa = 256, n_embd_v_gqa = 256
llama_kv_cache_init:        CPU KV buffer size =    44.00 MiB
llama_init_from_model: KV self size  =   44.00 MiB, K (f16):   22.00 MiB, V (f16):   22.00 MiB
llama_init_from_model:        CPU  output buffer size =     0.12 MiB
llama_init_from_model:        CPU compute buffer size =   148.01 MiB
llama_init_from_model: graph nodes  = 710
llama_init_from_model: graph splits = 1
CPU : SSE3 = 1 | SSSE3 = 1 | AVX = 1 | AVX2 = 1 | F16C = 1 | FMA = 1 | LLAMAFILE = 1 | OPENMP = 1 |
AARCH64_REPACK = 1 |
Model metadata: {'mradermacher.quantized_on': 'rich1', 'mradermacher.quantized_by': 'mradermacher',
'mradermacher.quantize_version': '2', 'general.url': 'https://huggingface.co/mradermacher/TinyLlama-R1-
LIMO-GGUF', 'general.quantization_version': '2', 'tokenizer.chat_template': "{% for message in messages
%}\n{% if message['role'] == 'user' %}\n{{ '<|user|>\n' + message['content'] + eos_token }}\n{% elif
message['role'] == 'system' %}\n{{ '<|system|>\n' + message['content'] + eos_token }}\n{% elif
message['role'] == 'assistant' %}\n{{ '<|assistant|>\n'  + message['content'] + eos_token }}\n{% endif
%}\n{% if loop.last and add_generation_prompt %}\n{{ '<|assistant|>' }}\n{% endif %}\n{% endfor %}",
'tokenizer.ggml.add_eos_token': 'false', 'tokenizer.ggml.padding_token_id': '0',
'general.dataset.0.name': 'LIMO', 'general.dataset.count': '1', 'general.base_model.0.repo_url':
'https://huggingface.co/Josephgflowers/Tinyllama-STEM-Cinder-Agent-v1', 'general.license': 'mit',
'llama.attention.value_length': '64', 'general.file_type': '14', 'general.dataset.0.organization':
'GAIR', 'tokenizer.ggml.add_bos_token': 'true', 'general.size_label': '1.1B', 'general.type': 'model',
'general.base_model.0.version': 'v1', 'llama.attention.head_count_kv': '4', 'general.base_model.0.name':
'Tinyllama STEM Cinder Agent v1', 'tokenizer.ggml.add_space_prefix': 'false',
'llama.rope.dimension_count': '64', 'tokenizer.ggml.bos_token_id': '1', 'general.name': 'Model Limo R3',
'llama.context_length': '8192', 'mradermacher.quantized_at': '2025-02-12T03:41:15+01:00',
'llama.embedding_length': '2048', 'llama.rope.scaling.type': 'linear', 'general.architecture': 'llama',
'mradermacher.convert_type': 'hf', 'general.source.url':
'https://huggingface.co/Josephgflowers/TinyLlama-R1-LIMO', 'general.base_model.count': '1',
'general.base_model.0.organization': 'Josephgflowers', 'llama.feed_forward_length': '5632',
'llama.rope.freq_base': '10000.000000', 'tokenizer.ggml.unknown_token_id': '0',
'tokenizer.ggml.eos_token_id': '2', 'general.dataset.0.repo_url': 'https://huggingface.co/GAIR/LIMO',
'llama.attention.layer_norm_rms_epsilon': '0.000010', 'llama.block_count': '22',
'llama.attention.head_count': '32', 'llama.attention.key_length': '64', 'tokenizer.ggml.pre': 'default',
'llama.vocab_size': '32000', 'llama.rope.scaling.factor': '4.000000', 'tokenizer.ggml.model': 'llama'}
Available chat formats from metadata: chat_template.default
Using gguf chat template: {% for message in messages %}
{% if message['role'] == 'user' %}
{{ '<|user|>
' + message['content'] + eos_token }}
{% elif message['role'] == 'system' %}
{{ '<|system|>
' + message['content'] + eos_token }}
{% elif message['role'] == 'assistant' %}
{{ '<|assistant|>
'  + message['content'] + eos_token }}
{% endif %}
{% if loop.last and add_generation_prompt %}
{{ '<|assistant|>' }}
{% endif %}
{% endfor %}
Using chat eos_token: </s>
Using chat bos_token: <s>
Iteration 1:
Explored Thought: <think>
Explored Thought: Okay, so I'm trying to solve this logic problem. Let me start by understanding the
problem first.
Iteration 2:
llama_perf_context_print:        load time =    3138.68 ms
llama_perf_context_print: prompt eval time =    3138.30 ms /   458 tokens (    6.85 ms per token,
145.94 tokens per second)
llama_perf_context_print:        eval time =    2317.40 ms /    99 runs   (   23.41 ms per token,
42.72 tokens per second)
llama_perf_context_print:       total time =    5494.29 ms /   557 tokens
Llama.generate: 26 prefix-match hit, remaining 28 prompt tokens to eval
llama_perf_context_print:        load time =    3138.68 ms
llama_perf_context_print: prompt eval time =     157.50 ms /    28 tokens (    5.62 ms per token,
177.78 tokens per second)
llama_perf_context_print:        eval time =    2154.65 ms /    99 runs   (   21.76 ms per token,
45.95 tokens per second)
llama_perf_context_print:       total time =    2350.46 ms /   127 tokens
Llama.generate: 26 prefix-match hit, remaining 50 prompt tokens to eval
llama_perf_context_print:        load time =    3138.68 ms
llama_perf_context_print: prompt eval time =     327.60 ms /    50 tokens (    6.55 ms per token,
152.63 tokens per second)
llama_perf_context_print:        eval time =    2119.51 ms /    99 runs   (   21.41 ms per token,
46.71 tokens per second)
llama_perf_context_print:       total time =    2486.77 ms /   149 tokens
Llama.generate: 26 prefix-match hit, remaining 28 prompt tokens to eval
Explored Thought: <think>
Explored Thought: Okay, so I need to come up with two concise next thoughts that evolve the original
idea. Let me start by thinking about what the original idea is. Hmm, I don't have the exact idea in mind
right now. Maybe it's something like "The importance of education in fostering lifelong learning and
personal growth." That sounds familiar. So, the original idea is about education being a tool for
personal and intellectual development.
Explored Thought: <think>
Explored Thought: Okay, so I'm trying to solve this logic problem, and I need to start by understanding
the current thought first. Let me break it down. The problem is a bit vague, but I think the main goal
is to apply the right logical principles to figure out the answer. Let me think through this step by
step.
Iteration 3:
llama_perf_context_print:        load time =    3138.68 ms
llama_perf_context_print: prompt eval time =     163.27 ms /    28 tokens (    5.83 ms per token,
171.50 tokens per second)
llama_perf_context_print:        eval time =    2539.86 ms /    99 runs   (   25.66 ms per token,
38.98 tokens per second)
llama_perf_context_print:       total time =    2742.70 ms /   127 tokens
Llama.generate: 26 prefix-match hit, remaining 116 prompt tokens to eval
llama_perf_context_print:        load time =    3138.68 ms
llama_perf_context_print: prompt eval time =     697.13 ms /   116 tokens (    6.01 ms per token,
166.40 tokens per second)
llama_perf_context_print:        eval time =    2243.09 ms /    99 runs   (   22.66 ms per token,
44.14 tokens per second)
```

```
llama_perf_context_print:       total time =    2979.92 ms /   215 tokens
Llama.generate: 26 prefix-match hit, remaining 28 prompt tokens to eval
llama_perf_context_print:        load time =    3138.68 ms
llama_perf_context_print: prompt eval time =     164.29 ms /    28 tokens (    5.87 ms per token,
170.43 tokens per second)
llama_perf_context_print:        eval time =    2176.97 ms /    99 runs   (   21.99 ms per token,
45.48 tokens per second)
llama_perf_context_print:       total time =    2380.81 ms /   127 tokens
Llama.generate: 26 prefix-match hit, remaining 94 prompt tokens to eval
Explored Thought: <think>
Explored Thought: Okay, so I need to come up with two concise next thoughts that evolve the idea
further. Let me start by understanding the current thought first. The user mentioned something about a
current thought being 'think', and they want to evolve it further.
Explored Thought: <think>
Explored Thought: Okay, so I need to come up with two concise next thoughts that evolve the original
idea about education fostering personal and intellectual growth. Let me start by thinking about what the
original idea is. The user mentioned that the original idea is about education being a tool for personal
and intellectual growth. That seems to align with what I know, but maybe there are other angles or
points to consider.
Explored Thought: <think>
Explored Thought: Okay, so I need to come up with two concise next thoughts that evolve the idea of
"Current Thinking" into a better or clearer thought. Let me start by understanding the original idea.
The user mentioned something about 'Current Thinking' being the current state of knowledge or
understanding. So maybe they want to explain what 'Current Thinking' is, or perhaps they want to break
it down into more specific areas.
Explored Thought: <think>
Explored Thought: Okay, so I'm trying to solve this logic problem, and I need to start by understanding
the current thought first. Let me break it down. The problem is a bit vague, but I think the main goal
is to apply the right logical principles to figure out the answer. Let me try to think through this step
by step.
====================================================================================================
Final Tree of Thoughts:
- You are an expert in formal logic and deductive reasoning. Below, you are presented with a complex
problem that requires multiple steps of logical inference. You must answer clearly, justifying each step
of your reasoning.

Problem:** Problem:** Problem:** Problem:** Problem:** Problem:** Problem:** Problem:** Problem:**
Problem
In a city, there are three types of people:
- **Truthful**: They always tell the truth.
- Liars**: They always lie.
- Random**: They respond randomly.

You come across three people: A, B and C. You know that each belongs to a different type (one is
truthful, one is a liar and one is random), but you don't know who is who.

You ask the following questions:

1. statement to A: ‘Are you the truthful one?’ - Answer, ‘Yes.’
2. Assertion to B: ‘Would A say you are the random one?’ - Answer, ‘No.’
3. Statement to C: ‘If I ask you if B is the liar, what would you answer?’ - Answer, ‘Yes.’

**Assignments:** 1.
1. Identify who is the truthful one, who is the liar, and who is the random one.
2. Explain step by step how you came to that conclusion.
3. Consider any ambiguity or cheating in the answers.

**Additional rules
- Reason as if the model could be flawed logically, pointing out any possible flaws or paradoxes.
- Do not make unwarranted assumptions.
- If there is more than one possible answer, analyse and evaluate each with its logical probability.

Translated with DeepL.com (free version)
  - <think>
    - <think>
      - <think>
      - Okay, so I need to come up with two concise next thoughts that evolve the idea further. Let me
start by understanding the current thought first. The user mentioned something about a current thought
being 'think', and they want to evolve it further.
    - Okay, so I need to come up with two concise next thoughts that evolve the original idea. Let me
start by thinking about what the original idea is. Hmm, I don't have the exact idea in mind right now.
Maybe it's something like "The importance of education in fostering lifelong learning and personal
```

```
growth." That sounds familiar. So, the original idea is about education being a tool for personal and
intellectual development.
      - <think>
      - Okay, so I need to come up with two concise next thoughts that evolve the original idea about
education fostering personal and intellectual growth. Let me start by thinking about what the original
idea is. The user mentioned that the original idea is about education being a tool for personal and
intellectual growth. That seems to align with what I know, but maybe there are other angles or points to
consider.
  - Okay, so I'm trying to solve this logic problem. Let me start by understanding the problem first.
    - <think>
      - <think>
      - Okay, so I need to come up with two concise next thoughts that evolve the idea of "Current
Thinking" into a better or clearer thought. Let me start by understanding the original idea. The user
mentioned something about 'Current Thinking' being the current state of knowledge or understanding. So
maybe they want to explain what 'Current Thinking' is, or perhaps they want to break it down into more
specific areas.
    - Okay, so I'm trying to solve this logic problem, and I need to start by understanding the current
thought first. Let me break it down. The problem is a bit vague, but I think the main goal is to apply
the right logical principles to figure out the answer. Let me think through this step by step.
      - <think>
      - Okay, so I'm trying to solve this logic problem, and I need to start by understanding the
current thought first. Let me break it down. The problem is a bit vague, but I think the main goal is to
apply the right logical principles to figure out the answer. Let me try to think through this step by
step.
Cost of time:  21.307671070098877
llama_perf_context_print:        load time =    3138.68 ms
llama_perf_context_print: prompt eval time =     596.01 ms /    94 tokens (    6.34 ms per token,
157.72 tokens per second)
llama_perf_context_print:        eval time =    2157.50 ms /    99 runs   (   21.79 ms per token,
45.89 tokens per second)
llama_perf_context_print:       total time =    2794.01 ms /   193 tokens

Process finished with exit code 0
```