

END LAB EXAM (END-TERM EXAM):

HALLTICKET- NO: 2403A51410

NAME: AMGOTH VIKAS NAYAK

BATCH: 24BTCAIAIB16

COURSE: AI ASSISTED CODING (AIAC)

SUBSET – 8:

Q1:

Design inverted index for search.

- Task 1: Use AI to implement index builder.
- Task 2: Add support for tokenization and stop words.

PROMPT:

```
ENDLABEXAM.py
-----
1 from collections import defaultdict
2 import nltk
3 from nltk.tokenize import word_tokenize
4 from nltk.corpus import stopwords
5
6 # Download necessary resources
7 nltk.download('punkt')
8 nltk.download('punkt_tab')
9 nltk.download('stopwords')
10
11 # Sample documents
12 documents = [
13     "The cat sat on the mat.",
14     "The dog barked at the cat.",
15     "The mouse ran away from the cat and dog."
16 ]
17
18 # Get NLTK's English stopwords
19 stop_words = set(str(stopwords.words('english')))
20
21 def clean_text(text):
22     # Lowercase and remove punctuation
23     import re
24     text = re.sub(r'[^\w\s]', '', text)
25     return text.lower()
26
27 def build_inverted_index(docs, stop_words):
28     inverted_index = defaultdict(list)
29     for doc_id, text in enumerate(docs):
30         tokens = word_tokenize(clean_text(text))
31         for token in tokens:
32             if token not in stop_words and doc_id not in inverted_index[token]:
33                 inverted_index[token].append(doc_id)
34     return inverted_index
35
36 index = build_inverted_index(documents, stop_words)
37 # Print the inverted index (sorted alphabetically)
38 for term in sorted(index.keys()):
39     print(f'{term}: {index[term]}')

Solve error in ENDLABEXAM.py
-----
But my output looks different something like ran away
ran away
Sorting the output alphabetically so it's clearer and easier
to read:
ENDLABEXAM.py +3 -3
print(f'{term}: {doc_ids}')
# Print the inverted index (sorted alphabetically)
for term in sorted(index.keys()):
    print(f'{term}: {index[term]}')


Running the script to verify the sorted output:
Auto-Ran command: python
python ENDLABEXAM.py
mouse: [2]
ran: [2]
sat: [0]

Output is now sorted alphabetically. Each word appears
once with its document IDs:
* away: [2] (document 2)
* barked: [1] (document 1)
* cat: [0, 1, 2] (documents 0, 1, and 2)
* dog: [1, 2] (documents 1 and 2)
* mat: [0] (document 0)
* mouse: [2] (document 2)
* ran: [2] (document 2)
* sat: [0] (document 0)

'ran' and 'away' are separate words, so they appear as
Plan, @ for context, / for commands
@ Agent Auto
```

CODE:

```

❶ ENDLABEXAM.py ●
❷ ENDLABEXAM.py > ...
1  from collections import defaultdict
2  import nltk
3  from nltk.tokenize import word_tokenize
4  from nltk.corpus import stopwords
5
6  # Download necessary resources
7  nltk.download('punkt')
8  nltk.download('punkt_tab')
9  nltk.download('stopwords')
10
11 # Sample documents
12 documents = [
13     "The cat sat on the mat.",
14     "The dog barked at the cat.",
15     "The mouse ran away from the cat and dog."
16 ]
17
18 # Get NLTK's English stopwords
19 stop_words = set(str(stopwords.words('english')))
20
21 def clean_text(text):
22     # Lowercase and remove punctuation
23     import re
24     text = re.sub(r'[^w\s]', '', text)
25     return text.lower()
26
27 def build_inverted_index(docs, stop_words):
28     inverted_index = defaultdict(list)
29     for doc_id, text in enumerate(docs):
30         tokens = word_tokenize(clean_text(text))
31         for token in tokens:
32             if token not in stop_words and doc_id not in inverted_index[token]:
33                 inverted_index[token].append(doc_id)
34
35     return inverted_index
36
37 index = build_inverted_index(documents, stop_words)
38 # Print the inverted index (sorted alphabetical)
39 for term in sorted(index.keys()):
40     print(f'{term}: {index[term]}')

```

OUTPUT:

```

Problems Output Debug Console Terminal Ports
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> & C:/Users/vikas/AppData/Local/Programs/Python/Python312/python.exe c:/Users/vikas/on
eDrive/Desktop/VikasAIAC/ENDLABEXAM.py
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\vikas\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]   C:\Users\vikas\AppData\Roaming\nltk_data...
[nltk_data] Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]   C:\Users\vikas\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
away: [2]
barked: [1]
cat: [0, 1, 2]
dog: [1, 2]
mat: [0]
mouse: [2]
ran: [2]
sat: [0]
o PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>

```

Q2:

Implement LRU cache for product pages.

- Task 1: Use AI to scaffold LRU implementation.
- Task 2: Add metrics and eviction tests.

PROMPT:

```
'''LRU Cache for Product Pages with Metrics and Tests'''
from typing import Optional, Dict, Any
from dataclasses import dataclass
from time import time
import unittest

@dataclass
class CacheMetrics:
    hits: int = 0
    misses: int = 0
    evictions: int = 0
    total_requests: int = 0
    cache_size: int = 0
    max_size: int = 0
    @property
    def hit_rate(self) -> float:
        return self.hits / self.total_requests * 100 if self.total_requests else 0.0
    @property
    def miss_rate(self) -> float:
        return self.misses / self.total_requests * 100 if self.total_requests else 0.0

class LRUNode:
    def __init__(self, key: str, value: Any):
        self.key, self.value = key, value
        self.prev = self.next = None
        self.timestamp = time()

class LRUCache:
    def __init__(self, capacity: int = 100):
        if capacity <= 0:
            raise ValueError("Capacity must be greater than 0")
        self.capacity = capacity
        self.cache: Dict[str, LRUNode] = {}
        self.metrics = CacheMetrics(max_size=capacity)
        self.head = LRUNode("", None)
        self.tail = LRUNode("", None)
        self.head.next = self.tail
        self.tail.prev = self.head
```

Reviewing the workspace to understand the project structure.
Explored 1 directory 2 files
Implementing an LRU cache for product pages with metrics and eviction tests. Starting with the LRU cache implementation:
To-dos
Implement LRU-cache-class with doubly-linked-list and hash map for O(1) operations
Add metrics tracking (hits, misses, evictions, cache size, etc.)
Create comprehensive test suite with eviction tests
Adding tests with metrics and eviction tests:
Started to-do Create comprehensive test suite with eviction tests
test_lru_cache.py (new) +478 -1
Comprehensive test suite for LRU Cache implementation. Includes metrics tracking and eviction tests

CODE:

Agents Editor File Edit Selection View Go Run ... Q VikasAIAC

ENDLABEXAM2.py X

ENDLABEXAM2.py -

```

1  """LRU Cache for Product Pages with Metrics and Tests"""
2  from typing import Optional, Dict, Any
3  from dataclasses import dataclass
4  from time import time
5  import unittest
6
7  @dataclass
8  class CacheMetrics:
9      hits: int = 0
10     misses: int = 0
11     evictions: int = 0
12     total_requests: int = 0
13     cache_size: int = 0
14     max_size: int = 0
15     @property
16     def hit_rate(self) -> float:
17         return (self.hits / self.total_requests * 100) if self.total_requests else 0.0
18     @property
19     def miss_rate(self) -> float:
20         return (self.misses / self.total_requests * 100) if self.total_requests else 0.0
21
22 class LRUNode:
23     def __init__(self, key: str, value: Any):
24         self.key, self.value = key, value
25         self.prev = self.next = None
26         self.timestamp = time()
27
28 class LRUCache:
29     def __init__(self, capacity: int = 100):
30         if capacity <= 0:
31             raise ValueError("capacity must be greater than 0")
32         self.capacity = capacity
33         self.cache: Dict[str, LRUNode] = {}
34         self.metrics = CacheMetrics(max_size=capacity)
35         self.head = LRUNode("", None)
36         self.tail = LRUNode("", None)
37         self.head.next = self.tail
38         self.tail.prev = self.head
39     def add_node(self, node: LRUNode):
40
41     class LRUCache:
42         def add_node(self, node: LRUNode):
43             self.head.next.prev = node
44             self.head.next = node
45             def _remove_node(self, node: LRUNode):
46                 node.prev.next, node.next.prev = node.next, node.prev
47             def _move_to_head(self, node: LRUNode):
48                 self._remove_node(node)
49                 self.add_node(node)
50             def _pop_tail(self) -> Optional[LRUNode]:
51                 last = self.tail.prev
52                 if last == self.head:
53                     return None
54                 self._remove_node(last)
55                 return last
56             def get(self, key: str) -> Optional[Any]:
57                 self.metrics.total_requests += 1
58                 node = self.cache.get(key)
59                 if node is None:
60                     self.metrics.misses += 1
61                     return None
62                 self._move_to_head(node)
63                 self.metrics.hits += 1
64                 return node.value
65             def put(self, key: str, value: Any) -> bool:
66                 node = self.cache.get(key)
67                 if node is None:
68                     new_node = LRUNode(key, value)
69                     if len(self.cache) >= self.capacity:
70                         tail = self._pop_tail()
71                         if tail:
72                             del self.cache[tail.key]
73                             self.metrics.evictions += 1
74                         self.cache[key] = new_node
75                         self._add_node(new_node)
76                         self.metrics.cache_size = len(self.cache)
77                         return True
78                     else:
79                         node.value, node.timestamp = value, time()

```

Review next file >

Cursor Tab Ln 264, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 64-bit

Agents Editor File Edit Selection View Go Run ...

Q. VikasAIAC

```

    ◆ ENDLABEXAM2.py x
    ◆ ENDLABEXAM2.py > ...
28     class LRUCache:
29         def put(self, key: str, value: Any) -> bool:
30             self._move_to_head(node)
31             return False
32         def delete(self, key: str) -> bool:
33             node = self.cache.get(key)
34             if node is None:
35                 return False
36             self._remove_node(node)
37             del self.cache[key]
38             self.metrics.cache_size = len(self.cache)
39             return True
40         def clear(self):
41             self.cache.clear()
42             self.head.next, self.tail.prev = self.tail, self.head
43             self.metrics.cache_size = 0
44         def size(self) -> int:
45             return len(self.cache)
46         def get_metrics(self) -> CacheMetrics:
47             self.metrics.cache_size = len(self.cache)
48             return self.metrics
49
50     class ProductPageCache:
51         def __init__(self, capacity: int = 100):
52             self.cache = LRUCache(capacity)
53         def get_product(self, product_id: str) -> Optional[Dict[str, Any]]:
54             return self.cache.get(product_id)
55         def cache_product(self, product_id: str, product_data: Dict[str, Any]) -> bool:
56             return self.cache.put(product_id, product_data)
57         def get_stats(self) -> Dict[str, Any]:
58             m = self.cache.get_metrics()
59             return {
60                 "size": m.cache_size,
61                 "capacity": m.max_size,
62                 "hits": m.hits,
63                 "misses": m.misses,
64                 "evictions": m.evictions,
65                 "total_requests": m.total_requests,
66                 "hit_rate": f'{m.hit_rate:.2f}%',
67                 "miss_rate": f'{m.miss_rate:.2f}%'
68             }
69
70     class TestLRUCache(unittest.TestCase):
71         def setup(self):
72             self.cache = LRUCache(capacity=1)

```

Review next file >

Cursor Tab Ln 264, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 64-bit

Agents Editor File Edit Selection View Go Run ...

Q. VikasAIAC

```

    ◆ ENDLABEXAM2.py x
    ◆ ENDLABEXAM2.py > ...
112     class TestLRUCache(unittest.TestCase):
113
114         def test_basic_ops(self):
115             self.cache.put("k1", "v1")
116             self.cache.put("k2", "v2")
117             self.assertEqual(self.cache.get("k1"), "v1")
118             self.assertIsNone(self.cache.get("k3"))
119
120         def test_eviction(self):
121             for i in range(4):
122                 self.cache.put(f"K{i}", f"V{i}")
123             self.assertIsNone(self.cache.get("k0"))
124             self.assertEqual(self.cache.get("k3"), "v3")
125
126         def test_lru_order(self):
127             self.cache.put("k1", "v1")
128             self.cache.put("k2", "v2")
129             self.cache.put("k3", "v3")
130             self.cache.get("k1")
131             self.cache.put("k4", "v4")
132             self.assertEqual(self.cache.get("k2"), "v1")
133             self.assertEqual(self.cache.get("k1"), "v1")
134
135         def test_update_no_eviction(self):
136             for i in range(3):
137                 self.cache.put(f"K{i}", f"V{i}")
138             self.cache.put("k0", "updated")
139             self.assertEqual(self.cache.size(), 3)
140             self.assertEqual(self.cache.get("k0"), "updated")
141
142         def test_delete_clear(self):
143             self.cache.put("k1", "v1")
144             self.cache.put("k2", "v2")
145             self.assertTrue(self.cache.delete("k1"))
146             self.cache.clear()
147             self.assertEqual(self.cache.size(), 0)

```

Review next file >

Cursor Tab Ln 264, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 64-bit

Q. VikasAIAC

```

Agents Editor File Edit Selection View Go Run ... ENDLABEXAM2.py x
VIKASAIAC
DFS.PY
ELAB.PY
ENDLABEXAM.PY
ENDLABEXAM2.PY
LAB ASSIGNMENT-6...
LAB-1-1.py
LAB-3-3.py
LAB-3-4.py
LAB-3-5.py
LAB-3-6.py
LAB-3T-2.py
LAB-3.py
LAB-11-9.PY
LAB11-1.py
LABEXAM-1.py
LABEXAM-2-1.py
LABEXAM-2-2.py
LABEXAM-3D0C.pdf
LABEXAM1-2.PY
LABEXAM3.py
LABEXAMSQ2.PY
LABEXAM4.py
IRU_CORE.py
main.py
requirements.txt
run_env_monitor_test...
sdk-snippets.md
SELLER_API_DOCS.md
seller-api-openapi.yaml
users.txt
zg2.py

OUTLINE
TIMELINE
0 △ 0

```

```

146 class TestMetrics(unittest.TestCase):
147     def test_hit_miss(self):
148         self.cache.get("k1")
149         self.cache.get("k2")
150         m = self.cache.get_metrics()
151         self.assertEqual(m.hits, 1)
152         self.assertEqual(m.misses, 1)
153         self.assertAlmostEqual(m.hit_rate, 50.0, places=1)
154
155     def test_eviction_count(self):
156         for i in range(4):
157             self.cache.put(f"v{i}", f"v{i}")
158         self.assertEqual(self.cache.get_metrics().evictions, 1)
159         self.cache.put("k4", "v4")
160         self.assertEqual(self.cache.get_metrics().evictions, 2)
161
162 class TestEvictionScenarios(unittest.TestCase):
163     def test_sequential_access(self):
164         c = LRUCache(capacity=3)
165         for i in range(1, 4):
166             c.put(f"p{i}", {"name": f"Product {i}"})
167             c.get(f"p{i}")
168             c.put("p4", {"name": "Product 4"})
169             c.get("p1")
170             self.assertIsNone(c.get("p1"))
171             self.assertIsNotNone(c.get("p4"))
172
173     def test_repeated_access(self):
174         c = LRUCache(capacity=3)
175         for i in range(1, 4):
176             c.put(f"p{i}", {"name": f"Product {i}"})
177             for _ in range(5):
178                 c.get("p1")
179                 c.get("p4", {"name": "Product 4"})
180                 self.assertIsNone(c.get("p2"))
181                 self.assertIsNotNone(c.get("p4"))
182
183     def test_update_eviction(self):
184         c = LRUCache(capacity=3)
185         for i in range(1, 4):
186             c.put(f"p{i}", {"price": i * 10})
187             c.nut("p1", {"price": 151})

```

Review next file >

Cursor Tab Ln 264, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 64-bit

Q. VikasAIAC

```

Agents Editor File Edit Selection View Go Run ... ENDLABEXAM2.py x
VIKASAIAC
DFS.PY
ELAB.PY
ENDLABEXAM.PY
ENDLABEXAM2.PY
LAB ASSIGNMENT-6...
LAB-1-1.py
LAB-3-3.py
LAB-3-4.py
LAB-3-5.py
LAB-3-6.py
LAB-3T-2.py
LAB-3.py
LAB-11-9.PY
LAB11-1.py
LABEXAM-1.py
LABEXAM-2-1.py
LABEXAM-2-2.py
LABEXAM-3D0C.pdf
LABEXAM1-2.PY
LABEXAM3.py
LABEXAMSQ2.PY
LABEXAM4.py
IRU_CORE.py
main.py
requirements.txt
run_env_monitor_test...
sdk-snippets.md
SELLER_API_DOCS.md
seller-api-openapi.yaml
users.txt
zg2.py

OUTLINE
TIMELINE
0 △ 0

```

```

164 class TestEvictionScenarios(unittest.TestCase):
165     def test_update_eviction(self):
166         c.put("p1", {"price": 15})
167         c.put("p4", {"price": 40})
168         self.assertIsNone(c.get("p2"))
169         self.assertIsNotNone(c.get("p1"))
170
171     class TestProductPageCache(unittest.TestCase):
172         def setUp(self):
173             self.cache = ProductPageCache(capacity=3)
174
175         def test_cache_product(self):
176             self.cache.cache_product("p1", {"title": "Laptop", "price": 999.99})
177             p = self.cache.get_product("p1")
178             self.assertDictEqual(p["title"], "Laptop")
179             self.assertDictEqual(p["price"], 999.99)
180
181         def test_stats(self):
182             self.cache.cache_product("p1", {"title": "P1"})
183             self.cache.get_product("p1")
184             self.cache.get_product("p2")
185             s = self.cache.get_stats()
186             self.assertEqual(s["hits"], 1)
187             self.assertEqual(s["misses"], 1)
188
189         def test_product_eviction(self):
190             for i in range(1, 4):
191                 self.cache.cache_product(f"p{i}", {"title": f"Product {i}"})
192                 self.cache.get_product("p1")
193                 self.cache.get_product("p2")
194                 self.cache.get_product("p3")
195                 self.cache.get_product("p4", {"title": "Product 4"})
196                 self.assertIsNone(self.cache.get_product("p2"))
197
198     class TestEdgeCases(unittest.TestCase):
199         def test_invalid_capacity(self):
200             with self.assertRaises(ValueError):
201                 LRUCache(capacity=0)
202             with self.assertRaises(ValueError):
203                 LRUCache(capacity=-1)
204
205         def test_single_capacity(self):
206             c = LRUCache(capacity=1)
207             c.put("k1", "v1")
208             c.nut("k2", "v2")

```

Review next file >

Cursor Tab Ln 264, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 64-bit

Q. VikasAIAC

```

Agents Editor File Edit Selection View Go Run ... ENDLABEXAM2.py x
VIKASAIAC
DFS.PY
ELAB.PY
ENDLABEXAM.PY
ENDLABEXAM2.PY
LAB ASSIGNMENT-6...
LAB-1-1.py
LAB-3-3.py
LAB-3-4.py
LAB-3-5.py
LAB-3-6.py
LAB-3T-2.py
LAB-3.py
LAB-11-9.PY
LAB11-1.py
LABEXAM-1.py
LABEXAM-2-1.py
LABEXAM-2-2.py
LABEXAM-3DOC.pdf
LABEXAM1-2.PY
LABEXAM3.py
LABEXAM3Q2.PY
LABEXAM4.py
IRU_CORE.py
main.py
requirements.txt
run_env_monitor_test...
sdk-snippets.md
SELLER_API_DOCS.md
seller-api-openapi.yaml
users.txt
zg2.py

OUTLINE
TIMELINE
Review next file >
Cursor Tab Ln 264, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 64-bit

Agents Editor File Edit Selection View Go Run ... ENDLABEXAM2.py x
VIKASAIAC
DFS.PY
ELAB.PY
ENDLABEXAM.PY
ENDLABEXAM2.PY
LAB ASSIGNMENT-6...
LAB-1-1.py
LAB-3-3.py
LAB-3-4.py
LAB-3-5.py
LAB-3-6.py
LAB-3T-2.py
LAB-3.py
LAB-11-9.PY
LAB11-1.py
LABEXAM-1.py
LABEXAM-2-1.py
LABEXAM-2-2.py
LABEXAM-3DOC.pdf
LABEXAM1-2.PY
LABEXAM3.py
LABEXAM3Q2.PY
LABEXAM4.py
IRU_CORE.py
main.py
requirements.txt
run_env_monitor_test...
sdk-snippets.md
SELLER_API_DOCS.md
seller-api-openapi.yaml
users.txt
zg2.py

OUTLINE
TIMELINE
Review next file >
Cursor Tab Ln 264, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 64-bit

```

OUTPUT:

```
Problems Output Debug Console Terminal Ports Python + ×

test_lru_order (_main_.TestLRUCache.test_lru_order) ... ok
test_update_no_eviction (_main_.TestLRUCache.test_update_no_eviction) ... ok
test_eviction_count (_main_.TestMetrics.test_eviction_count) ... ok
test_hit_miss (_main_.TestMetrics.test_hit_miss) ... ok
test_cache_product (_main_.TestProductPageCache.test_cache_product) ... ok
test_product_eviction (_main_.TestProductPageCache.test_product_eviction) ... ok
test_stats (_main_.TestProductPageCache.test_stats) ... ok

-----
Ran 15 tests in 0.001s

OK

=====
Performance Test
=====

Cache Statistics:
  size: 100
  capacity: 100
  hits: 50
  misses: 50
  evictions: 100
  total_requests: 100
  hit_rate: 50.0%
  miss_rate: 50.00%

PS C:\Users\vikas\OneDrive\Desktop\WikasATAC> 
```

THANK YOU