In [ ]:
```python
import heapq
from collections import defaultdict

# Define compression functions
def compress_rle(text):
    compressed_text = []
    count = 0
    for i in range(1, len(text)):
        if text[i] == text[i - 1] and count < RUN_LENGTH:
            count += 1
        else:
            compressed_text.append((text[i - 1], count ))  # Length starts from 0
            count = 0
    # Append the last symbol with its count
    compressed_text.append((text[-1], count))
    return compressed_text
```

In [ ]:
```python
# Define decompression functions
def decompress_rle(compressed_text):
    decompressed_text = ''
    for symbol, length in compressed_text:
        decompressed_text += symbol * (length + 1)  # Increment length to match RLE
    return decompressed_text
```

In [ ]:
```python
RUN_BITS = 2
RUN_LENGTH = 2**RUN_BITS

text = "AAABBBCCDAA"
```

In [ ]:
```python
#Ex form of compressed text: [('A', 2), ('B', 2), ('C', 1), ('D', 0), ('A', 1)]
# Test the RLE compression and decompression

compressed_text = compress_rle(text)
print("Compressed text:", compressed_text)

decompressed_text = decompress_rle(compressed_text)
print("Decompressed text:", decompressed_text)
```

```
Compressed text: [('A', 2), ('A', 1), ('B', 2), ('C', 1), ('D', 0), ('A', 1)]
Decompressed text: AAAAABBBCCDAA
```