**Lab 6**

**Arithmetic coding**

**The purpose of the work** : to study the principles of arithmetic coding used for data compression and to develop an algorithm for coding and decoding text based on it. During the work, students will learn to evaluate the effectiveness of compression using arithmetic coding and compare it with other methods.

**Theoretical part**

Arithmetic coding is a data compression technique that converts an entire message into a single number between 0 and 1. Instead of encoding each character with a fixed number of bits, as other techniques do, arithmetic coding processes all the characters together, gradually refining the range for the encoded number.

**Basic principles:**

- Range initialization:

The initial interval for encoding is the segment [0, 1). The whole process consists of gradually narrowing this interval as each symbol is processed, based on the probability of the symbols appearing.

- Interval splitting:

The interval is divided proportionally to the probabilities of the symbols in the original message. The more frequently a symbol occurs, the larger the segment it occupies.

- Character encoding:

As each symbol is processed, the current interval is divided into smaller sections, and the subinterval corresponding to the symbol is selected.

- Final encoding:

As a result, the message is encoded by a single number, which belongs to the final subinterval .

**Example 1:**

Let us have a string "ABB" and the following character probabilities:

A: 0.5

B: 0.5

Initial interval: [0, 1).

After processing the first symbol (A), the interval is divided into two:

A: [0, 0.5)

B: [0.5, 1)

The next symbol (B) narrows the interval for B:

* A: [0.5, 0.75)
* B: [0.75, 1)

Thus, the string "ABB" is encoded as a number falling in the interval [0.75, 0.875).

**Example 2 (more detailed) for the string "AB AA B":**

1. **Determining the probabilities of symbols**

We have a string "AB AA B" that contains 7 symbols: *A* occurs 3 times. *B* occurs 2 times. *Space* occurs 2 times. The total length of the string is 7 symbols. Accordingly, the probabilities of the symbols are: P(A) = 3/7 ≈ 0.4286, P(B) = 2/7 ≈ 0.2857, P(space) = 2/7 ≈ 0.2857.

Now we divide the interval [0, 1) into proportional parts based on the probabilities:

Interval for A: [0.0, 0.4286)

Interval for B: [0.4286, 0.7143)

Space interval: [0.7143, 1.0)

1. **Encoding of the string "AB AA B":**

To encode a string, we will narrow the interval, each time updating its boundaries depending on the characters.

**Step 1 : Encode the character "A"**

The initial interval was: [0.0, 1.0). The interval for the character "A" = [0.0, 0.4286). So our new interval for the string = [0.0, 0.4286)

**Step 2 : Encode the character "B"**

The current interval, according to the previous step, is: [0.0, 0.4286). We divide this interval into parts according to the probabilities:

- Interval for character "A" = [0.0, 0.1837)

- Interval for character "B" = [0.1837, 0.3061)

- Space interval = [0.3061, 0.4286)

Since the current symbol is "B", we select the interval [0.1837, 0.3061).

**Step 3 : Encode the space**

The current interval is: [0.1837, 0.3061).

We divide this interval:

- Interval for "A" = [0.1837, 0.2326)

- Interval for "B" = [0.2326, 0.2704)

- Interval for space = [0.2704, 0.3061)

The current symbol is "space", so we select the interval [0.2704, 0.3061).

**Step 4 : Encode the character "A"**

Current interval: [0.2704, 0.3061).

We divide this interval:

- Interval for "A" = [0.2704, 0.2851)

- Interval for "B" = [0.2851, 0.2956)

- Interval for space = [0.2956, 0.3061)

Since the current Symbol is "A", we select the interval [0.2704, 0.2851).

**Step 5 : Encode the character "A"**

Current interval: [0.2704, 0.2851)

We divide this interval:

- Interval for "A" = [0.2704, 0.2765)

- Interval for "B" = [0.2765, 0.2807)

- Space interval = [0.2807, 0.2851)

Symbol - "A", select the interval [0.2704, 0.2765).

**Step 6 : Encode the space**

Current interval: [0.2704, 0.2765)

We divide this interval:

- Interval for "A" = [0.2704, 0.2729)

- Interval for "B" = [0.2729, 0.2749)

- Interval for space = [0.2749, 0.2765)

The current character is a space, we select the interval [0.2749, 0.2765).

**Step 7 : Encode the character "B"**

Current interval: [0.2749, 0.2765)

We divide this interval:

- Interval for "A" = [0.2749, 0.2755)

- Interval for "B" = [0.2755, 0.2760)

- Interval for space = [0.2760, 0.2765)

The current symbol is "B", we select the interval [0.2755, 0.2760).

**Final interval**

The encoded value will be ***any number*** in the range [0.2755, 0.2760).

For example, you can choose a number in the middle: 0.27575

The string "AB AA B" has been encoded into the number **0.27575** . This number is unique to the string and all the characters in it. The same narrowing process can be used to decode this number in reverse.

**Advantages:**

1. Arithmetic coding reaches almost the theoretical limit of compression.
2. The encoding is more efficient than, for example, the Huffman code, especially for long messages.

**Practical part**

**Objectives of the laboratory work:**

1. Implement an arithmetic coding algorithm for an arbitrary string.
2. Conduct testing of the program on various initial data. Formulate conclusions on which data arithmetic coding is more effective.
3. Test the program on different strings and probabilities. Compare the length of the compressed message with the original.
4. Calculate the compression ratio:

С ompress = V original / V compressed

1. To evaluate the efficiency of arithmetic coding for different data sets (text, random data).
2. Evaluate compression efficiency compared to other methods (e.g. RLE or Huffman coding). See previous labs.

***p.s. Before defending your work, manually calculate the code for the string " ABCABEAAFB " and compare the result with the software calculation.***

**Progress of work:**

1. **Program development:**
   * **Input data:** String to encode and frequency (probability) of occurrence of symbols.
   * **Coding algorithm:**
     + Initializing the interval [0, 1).
     + For each symbol:
       - Dividing the current interval according to probabilities.
       - Select a subinterval for a symbol.
     + Obtaining a final number representing the encoded message.