Week 11 Assignment

[(35, 5), (43, 8), (44. 7), (50, 2)]

1. To write the resulting VarInt compression for the given list, we follow these steps:

1. Convert each of the integers in the list to their binary representation.
2. For each binary representation, determine the number of bytes needed to represent the integer by counting the number of bits in the binary representation and dividing by 7 (since each byte can hold 7 bits of information).
3. For each integer, add a prefix to its binary representation, where the prefix is a byte with the most significant bit set to 1 and the other bits set to 0. The number of prefix bytes is equal to the number of bytes needed to represent the integer.
4. Concatenate the prefix bytes and the binary representation of each integer to get the resulting VarInt compression.

For example, for the first integer in the list (35), its binary representation is `100011`, which requires 2 bytes to represent (since it has 6 bits). The prefix for this integer would be `11000000`, which is 1 byte with the most significant bit set to 1 and the other bits set to 0. The resulting VarInt compression for this integer would be `11000000 100011`, which is 3 bytes in total.

Here is the resulting VarInt compression for the given list:

11000000 100011
11000001 101011
11000001 101111
11000010 110010

2. Simple-9 compression.

The first value in each tuple represents the integer to be compressed and the second value represents the number of times that integer appears in the list.

First, we will try to find the number of bits needed to represent the highest integer in the list, which is 50. To do this, we will take the log base 2 of 50 and round up to the nearest whole number. The result is 6.

Now, we will try to find the number of bits needed to represent the highest number of occurrences, which is 8. To do this, we will take the log base 2 of 8 and round up to the nearest whole number. The result is 3.

Using these values, we can determine that we will need a total of 9 bits for each entry in the Simple-9 compression. The first 6 bits will be used to represent the integer and the last 3 bits will be used to represent the number of occurrences.

Using this information, we can now create the Simple-9 compression for the given list:

(1, 000101, 000101) - This represents the tuple (35, 5) (1, 001011, 001000) - This represents the tuple (43, 8) (1, 001100, 000111) - This represents the tuple (44, 7) (1, 010010, 000010) - This represents the tuple (50, 2)

The resulting Simple-9 compression is: [1, 000101, 000101, 1, 001011, 001000, 1, 001100, 000111, 1, 010010, 000010]

3. Bit-Aligned Compression:

[(3, 5), (4, 3), (4, 4), (5, 0)]

Explanation:

For the first tuple (35, 5), the number 35 can be represented as 111011 in binary, which is 6 bits long. Since the second value in the tuple is 5, we know that the first 5 bits of the binary representation should be used. So the resulting compressed tuple is (3, 5), where 3 is the binary representation of the first 5 bits (011) and 5 is the number of bits used.

For the second tuple (43, 8), the number 43 can be represented as 101011 in binary, which is 6 bits long. Since the second value in the tuple is 8, we know that all 6 bits of the binary representation should be used. So the resulting compressed tuple is (4, 3), where 4 is the binary representation of all 6 bits (101011) and 3 is the number of bits used.

For the third tuple (44, 7), the number 44 can be represented as 101100 in binary, which is 6 bits long. Since the second value in the tuple is 7, we know that the first 7 bits of the binary representation should be used. So the resulting compressed tuple is (4, 4), where 4 is the binary representation of the first 7 bits (1011000) and 4 is the number of bits used.

For the fourth tuple (50, 2), the number 50 can be represented as 110010 in binary, which is 6 bits long. Since the second value in the tuple is 2, we know that only the first 2 bits of the binary representation should be used. So the resulting compressed tuple is (5, 0), where 5 is the binary representation of the first 2 bits (11) and 0 is the number of bits used.