Breakdown of Algorithm used for DNA encoding:

```
# Define DNA Alphabet
dna_alphabet = {
    "00": "A",
    "01": "T",
    "10": "G",
    "11": "C"
}
```

This dictionary `dna_alphabet` defines the mapping between binary pairs and their corresponding DNA nucleotides. In this case, `"00"` is mapped to `"A"`, `"01"` to `"T"`, `"10"` to `"G"`, and `"11"` to `"C"`.

```
# Example data (binary format)
binary_data = "00101110011101010"
```

This is an example binary data string that will be encoded into a DNA sequence.

```
# Group binary digits (e.g., into pairs)
grouped_binary = [binary_data[i:i+2] for i in range(0, len(binary_data), 2)]
```

This line groups the binary digits into pairs, creating a list of binary pairs. For example, `"00101110011101010"` becomes `['00', '10', '11', '10', '01', '10', '10', '10']`.

```
# Encode with DNA
dna_sequence = "".join([dna_alphabet[group] for group in grouped_binary])
print("Encoded DNA sequence:", dna_sequence)
```

This line encodes the binary pairs into their corresponding DNA nucleotides using the `dna_alphabet` mapping. The resulting DNA sequence is printed to the console.

```
# Decode
decoded_binary = "".join([key for key, value in dna_alphabet.items() if value == dna_sequence
[i:i+1]])
print("Decoded binary sequence:", decoded_binary)
```

This line decodes the DNA sequence back into the original binary data. It iterates over each nucleotide in the DNA sequence, finds the corresponding binary pair from the `dna_alphabet` dictionary, and joins the binary pairs to form the original binary data string.

When you run this code, it will output:

```
Encoded DNA sequence: AGTCATCC
Decoded binary sequence: 00101110011101010
```