Hash Function Analysis Report

Introduction

This report evaluates six different hash functions used in the Word Count Wizard. The results of each hash function are based on the number of collisions, the total number of unique words, and the total number of words when applied to the test dataset. The hash function with the fewest collisions is selected as the default.

Hash Functions and Pseudocode

Below are the pseudocode and corresponding C++ implementations of the six hash functions:

1. Summation Hash Code

Pseudocode:

Algorithm 1 SummationHashCode(key)

Input: String key
Output: Hash value
$hash \leftarrow 0$
for each character c in key **do**
    $hash \leftarrow hash + \text{ASCII}(c)$ ▷ Add ASCII value of character
return $hash$ mod capacity

C++ Code:

```
// Summation Hash Code
if (selected_hash_function == 1) {
    unsigned long hash = 0;
    for (char c : key) {
        hash += c; // Add ASCII value of each character
    }
    return hash % capacity;
}
```

2. Polynomial Hash Code

Pseudocode:

Algorithm 2 PolynomialHashCode(key)

Input: String key
Output: Hash value
$hash \leftarrow 0$
$prime \leftarrow 31$ ▷ A small prime number
for each character c in key **do**
    $hash \leftarrow (hash \cdot prime) + \text{ASCII}(c)$ ▷ Polynomial accumulation
return $hash$ mod capacity

C++ Code:

```
// Polynomial Hash Code
if (selected_hash_function == 2) {
    unsigned long hash = 0;
    unsigned long prime = 31; // A small prime number
    for (char c : key) {
        hash = (hash * prime) + c; // Polynomial accumulation
    }
    return hash % capacity;
}
```

3. DJB2 Hash Code

Pseudocode:

Algorithm 3 DJB2HashCode(key)

Input: String key
Output: Hash value
$hash \leftarrow 5381$ ▷ Initial value
for each character c in key **do**
    $hash \leftarrow ((hash \ll 5) + hash) + \text{ASCII}(c)$ ▷ hash × 33 + c
return $hash$ mod capacity

C++ Code:

```
// DJB2 Hash Code
if (selected_hash_function == 3) {
    unsigned long hash = 5381;
    for (char c : key) {
        hash = ((hash << 5) + hash) + c; // hash * 33 + c
    }
```

```
    return hash % capacity;
}
// Source: http://www.cse.yorku.ca/~oz/hash.html
```

4. SDBM Hash Code

Pseudocode:

Algorithm 4 SDBMHashCode(key)

Input: String key
Output: Hash value
$hash \leftarrow 0$
for each character c in key **do**
    $hash \leftarrow \text{ASCII}(c) + (hash \ll 6) + (hash \ll 16) - hash$
return $hash$ mod capacity

C++ Code:

```
// SDBM Hash Code
if (selected_hash_function == 4) {
    unsigned long hash = 0;
    for (char c : key) {
        hash = c + (hash << 6) + (hash << 16) - hash;
    }
    return hash % capacity;
}
// Source: https://www.partow.net/programming/hashfunctions/#SDBMHashFunction
```

5. Cycle Shift Hash Code

Pseudocode:

Algorithm 5 CycleShiftHashCode(key)

Input: String key
Output: Hash value
$hash \leftarrow 0$
for each character c in key **do**
    $hash \leftarrow (hash \ll 4) | (hash \gg 28)$ ▷ Rotate left by 4 bits
    $hash \leftarrow hash + \text{ASCII}(c)$
return $hash$ mod capacity

C++ Code:

```
// Cycle Shift Hash Code
if (selected_hash_function == 5) {
```

```
    unsigned long hash = 0;
    for (char c : key) {
        hash = (hash << 4) | (hash >> 28); // Rotate left by 4 bits
        hash += c;
    }
    return hash % capacity;
}
```

Performance Results

The performance results for each hash function when applied to the test dataset are summarized below:

1. **Summation Hash Code:**
   - The number of collisions is: 22411
   - The number of unique words is: 24139
   - The total number of words is: 306569
2. **Polynomial Hash Code:**
   - The number of collisions is: 584
   - The number of unique words is: 24139
   - The total number of words is: 306569
3. **DJB2 Hash Code:**
   - The number of collisions is: 666
   - The number of unique words is: 24139
   - The total number of words is: 306569
4. **SDBM Hash Code:**
   - The number of collisions is: 693
   - The number of unique words is: 24139
   - The total number of words is: 306569
5. **Cycle Shift Hash Code:**
   - The number of collisions is: 689
   - The number of unique words is: 24139
   - The total number of words is: 306569
6. **Length-Based Hash Code:**
   - The number of collisions is: 22402
   - The number of unique words is: 24139
   - The total number of words is: 306569

Conclusion

Based on the analysis of the six hash functions, the **Polynomial Hash Code** was found to produce the fewest collisions, with only 584 collisions during the test. As such, the Polynomial Hash Code has been selected as the default hash function for the Word Count Wizard. This choice ensures better performance and minimizes collision handling overhead compared to other hash functions.