1 System Configuration

The CPU used is an Intel i7-4578 @3.0GHz. RAM is 8GB, and cache size is 4MB.

2 Construction Time

1. String s1

   23 microseconds

2. String s2

   24 microseconds

3. Human BRCA2 gene

   177 microseconds

4. Tomato's chloroplast genome

   1,720 microseconds

5. Yeast's Chromosome 12

   11,370 microseconds

3 Justification

The performance statistics listed above certainly meet the expectations for performance. As the input size increased, the performance did as well. For example, for input 3, the base pair number to microsecond ratio was approximately 64. For input 4, the ratio was approximately 90, and for input 5, the ration was approximately 95. This demonstrates that as the input size increased, the number of base pairs processed per microsecond increased. Clearly, this suffix tree implementation operates in $O(n)$ time.

4 Implementation Constant

For every input byte, the suffix tree in theory uses 64 bytes. Each node in the tree is represented by a Node object, which contains four pointers (for navigation and siblings) and four integers (for node information) for a total of 32 bytes, and the number of nodes in the tree is bounded by $2n$, where n is the number of bytes. In reality, when compiled in 32-bit mode with compiler optimization, the code ran with a peak memory usage of 3.35MB for the Human BRCA2 gene, which contains 11,382 base pairs and consequently bytes. Of course, this statistic includes all program memory, not just the suffix tree.

5 BWT Index

BWT indeces have been included for the tomato genome and yeast chromosome.

6 Exact Matching Repeat

To find the longest exact matching repeat, all that is needed is to return the node ids (which are suffix ids) of the children of the lowest internal node as the starting indeces and the string depth of the lowest internal node as the length. To find the lowest internal node, the tree maintained a pointer to the current lowest internal node during construction. Whenever a new internal node was created, its string depth was compared to the previous lowest internal node. If it was deeper, the new node became the lowest internal node. The cost of finding the lowest internal node in this case is negligible, since it is simply setting a pointer during a specific function call (without the need for any checks). Reading the child suffix ids depends only on the input alphabet size for time complexity. The longest exact matching repeats for all inputs are as follows:

1. String s1

   Length: 3

   Starting at: 1 3

2. String s2

   Length: 4

   Starting at: 1 4

3. Human BRCA2 gene

   Length: 14

   Starting at: 6404 6164

4. Tomato's chloroplast genome

   Length: 48

Starting at: 88130 88150

5. Yeast's Chromosome 12

   Length: 8375

   Starting at: 460555 451418