**Computational Problem Formulation**

Let *Genome* be a string over alphabet “ACGT”. Let *s* be the list of unique symbols in *Genome*, where is the most frequent symbol of *Genome*, is the second most frequent symbol of *Genome*, etc. Let be the frequency (number of occurrences) of in *Genome*. Let *Cuts* be a set of indices in *Genome*, and let be a list of strings resulting from splitting *Genome* at the indices in *Cuts*.

Let

Input: A string *Genome*

Output: A set of indices *Cuts* in *Genome* such that is minimized

**DESCRITIONS OF 5 CASES**

* If , I only need the InfoByte (to tell me the “Huffman Tree structure”, i.e., that I only have one symbol + what symbol it is) and the 8 bytes representing the number of characters of the input, so 9 bytes total.
* If , I need the InfoByte (to tell me the Huffman Tree structure), 8 bytes representing the number of characters of the input, and the actual encoded message, where each symbol in the original message is represented by a single byte , so 9 bytes for info + for the message itself
* If , I need the InfoByte (to tell me the Huffman Tree structure), 8 bytes representing the number of characters of the input, and the actual encoded message. Notice that, in a Huffman Tree of 3 leaves, , , and . Therefore, the filesize is 9 bytes for info + for the message itself
* If , I need the InfoByte (to tell me the Huffman Tree structure), 8 bytes representing the number of characters of the input, but there are two topologies, and the specific topology affects the overall size:
  + If , the Huffman Tree will have the balanced topology, so each symbol will be represented by 2 bits: . Therefore, since each symbol is 2 bits (as opposed to the original 8), the filesize is 9 bytes for info + for the message itself
  + If , the Huffman Tree will have the unbalanced topology, so . Therefore, the filesize is 9 bytes for info +