Code Jam 2015 - World Finals

Analysis: Pretty Good Proportion

A brute force O($\mathbf{N}^2$) strategy will work for the Small dataset: for each position, compute the cumulative number of ones seen so far, and then check every possible pair of start and end points in the sequence. But **N** can be up to half a million in the Large dataset!

We will read the string from left to right and keep track of the number of ones O_i seen after reading the first i digits. We will examine each i between 0 and **N**, inclusive, so that we consider both the empty prefix and the full string. We will maintain an initially empty set of points, and for each i, we will add p_i = (i, **F** × i - O_i) to the set. Notice that the fraction of ones in the substring starting at position i+1 and ending at position j is (O_j - O_i) / (j - i), and the difference from **F** is **F** - (O_j - O_i) / (j - i) = (**F**(j - i) - O_j + O_i) / (j - i), which is the slope between p_i and p_j. Finding the p_i and p_j that minimize the absolute value of this slope is equivalent to solving the problem.

It is hard to see (but easy to prove) that the slope with minimal absolute value occurs between two points that are consecutive in sorted order by y coordinate: if there is a point r that has a y coordinate between the y coordinates of two other points p and q, the slope between p and q (call it s) is the weighted average of the slope between p and r and the slope between q and r; either those two slopes are equal to s, or one of them must be less than s. A more visual way to see this is to draw the segment p-q and place r on it. That makes all three slopes equal. Moving r horizontally clearly makes the slope of one of the segments pr and qr increase and the slope of the other one decrease.

This insight saves us from checking all pairs of points, and reduces the time complexity of the problem to O(**N** log **N**), which is imposed by the single sorting operation.