DECIPHERING THE MAYAN WRITING (Expected solution)

Author: Cesar Cepeda (Mexico)

The contestant should note that any solution that actually tries to match every possible permutation of *W* within *S* is doomed to failure. Even with a small alphabet the total number of permutations for *W* is very big. An optimal solution should take into account the fact that we are looking for *every* possible permutation of *W*, in other words, we don't care for the order of the characters.

OPTIMAL SOLUTION: The optimal solution uses a queue where it pushes consecutive characters of *S* until an error condition is reached. An error condition is reached whenever
- A character *c* that does not appear in *W* is in the queue.
- The number of times a character *c*, that appears in *W*, is in the queue exceeds the number of times that character appears in *W*.

After an error condition is reached, characters are popped from the queue until the error condition is cleared. In order to check for an error condition, the optimal solution keeps two vectors of length equal to the alphabet, each entry corresponds to a character in the alphabet. These two vectors maintain the number of times each character appears in *W* and the number of times each character is in the queue. Whenever you push or pop a character from the queue, the entry for that character must be checked in both vectors.

If at any time the length of the queue is equal to the length of *W* without having an error condition, the you have found a match.

Since every character in *S* is pushed and popped once into the queue the complexity of the resulting algorithm is *O(S)*.

SOLUTION 2: Instead of keeping two vectors to check for an error condition, every time a character is pushed or popped from the queue check every character in the alphabet to catch an error condition. This solution yields a complexity $O(S \Sigma)$ and should be able to score 70 points.

SOLUTION 3: Starting from every position in *S*, look for a match by checking the number of times each character appears in *W*, if an error is found, stop an advance to the next position in *S*. This solution can yield a worst case complexity of $O(S^2)$ and scored between 50 and 65 points depending on the implementation.

Test run #	Group/Value	g = |W|	|S|
1	1/0	4	11
2	2/50	3	100
3	2/50	6	102
4	2/50	8	209
5	2/50	10	500
6	2/50	10	516
7	2/50	10	300
8	2/50	8	200
9	2/50	8	995
10	3/5	50	10000
11	4/5	50	15000
12	5/5	50	50000
13	6/5	1000	1500000
14	7/30	2000	2000000
15	7/30	2500	2750000
16	7/30	3000	3000000