CS240, Spring 2022
Assignment 5: Question 6

Q6a) Construct the last occurrence function L for pattern $P = ramammam$ where $\Sigma = a, k, m, r$.

c	a	k	m	r
L(c)	6	-1	7	0

Q6b) Trace the search for P in $T = ramaamamkmamramammmam$ using the Boyer-Moore algorithm. Indicate in a table which characters of P were compared with which characters of T.

r	a	m	a	a	m	a	m	k	m	a	m	r	a	m	a	m	m	a	m
				m	m	a	m												
								m											
													m	m	a	m			
																a	m		
																		m	
												r	a	m	a	m	m	[a]	m

Q6c) For any $m \geq 1$ and any $n \geq m$, give a pattern P and a text T such that the Boyer-Moore algorithm looks at exactly $\Theta(n/m)$ characters.

For any $m \geq 1$, we will consider a pattern where the first $m-1$ characters are a and $p[m-1] = b$. For example when $m = 3$, the pattern we will be:

$$P = aab$$

The text will simply be n characters of a. For example when $n = 5$ we will have:

$$T = aaaaa$$

Note that the text will never contain b (its not in the alphabet), this means that every time we compare the pattern to the text we will use the bad character heuristic.

Since the last character will always not match, we will always move forward m characters. Thus for each subset of size m we will do one comparison. Our number of comparisons is given by:

$$\lceil \frac{n}{m} \rceil \text{ or } \theta(\frac{n}{m})$$

Q6d) For any $m \geq 1$ and any $n \geq m$ that is a multiple of m, give a pattern P and a text T such that the Boyer-Moore algorithm looks at all characters of the text at least once and returns with failure. Justify your answer.

For any $m \geq 1$, we will consider a pattern where its just b followed by a repeated m times For example when $m = 3$, the pattern we will be:

$$P = baa$$

Since n is divisible by m we will create the integer $i = \frac{n}{m}$, such that the text is made of i copies of S. In other words the text will look like:

$$i \times [aaaa...]$$

For example if m = 3 and n =9 our text would look like:

$$T = aaaaaaaaa$$

Note that the (m-1) suffix of S' will always match the pattern as they are both "a". However we know that P'[0] will always contain the "b" which is not in our text.

Because of this we would end up shifting the pattern over by 1 and comparing from the end again. Overall this means that since our pattern would shift once every time we will look at each character in the list at least one time.

Q6e) A number of heuristics can be used with Boyer-Moore to reduce the number of comparisons performed between P and T. Suppose we use Boyer-Moore with only the Peek heuristic. The Peek heuristic states that if $P[j] \neq T[i]$ and $P[j-1] \neq T[i-1]$ then the next location to search for P at is $T[i+m-1]$. Show that the Peek heuristic may fail to find P in T, i.e., find a pattern P, and a text T containing P,such that Peek fails to find P in T. Justify your answer in a short paragraph.

Consider the pattern:

$$P = \text{aabcdeaa}$$

And consider it with the following text.

$$T = \text{aa}\mathbf{aabcdeaa}\text{defgii}$$

From this we can see that T contains P. Using the Peek heuristic of Boyer-Moore, we will start by comparing the last two values in P to their corresponding value in T ($i = j = 7$).

$$P[i-1, i] = aa$$

$$T[j-1, j] = de$$

Because both $P[j] \neq T[i]$ and $P[j-1] \neq T[i-1]$ we will move forward in the text as we will compare P to $T[13] = f$ and $T[12] = e$ next. However, since this is also a mismatch (we are comparing them to aa) we will skip forward again, however we will run out of space so we must return a failure.

Therefore we have shown how using this heuristic with P = aabcdeaa and T = aaaabcdeaadefgii will fail to find the pattern in the text.