CSC721 Algorithms Fall 2017

Knuth-Morris-Pratt String Search Algorithm (from Wikipedia)

Inputs: Character string (array) S (text to be searched)
Character string (array) W (word to find)
Output: Index in S where W begins (if W occurs in S), $|S|$ otherwise.
Method:

```
m = 0   ;
i = 0   ;
while ( (m + i) < |S| ) do {
    if ( W[i] == S[m + i] ) {
        if ( i == (|W| - 1) ) return m   ;
        i = i + 1   ;
    }
    else { // Mismatch at position W[i] and S[m+i]
        m = m + i - T[i]   ;
        if ( T[i] > -1 )
            i = T[i]   ;
        else
            i = 0   ;
    }
}
return |S|   ;   // Search is unsuccessful.
```

Computing the Table T[i]

Definition:

$T[i]$ = length of the longest proper prefix of W which matches a suffix of the substring of W ending at position $i - 1$

Examples:

i	0	1	2	3	4	5	6
W[i]	A	B	C	D	A	B	D
T[i]	-1	0	0	0	0	1	2

i	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
W[i]	P	A	R	T	I	C	I	P	A	T	E		I	N		P	A	R	A	C	H	U	T	E
T[i]	-1	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	1	2	3	0	0	0	0	0

Note: We need to consider suffixes of a size $\ell + 1$ only if a valid suffix of size ℓ was found at the previous stage.

Knuth-Morris-Pratt Table

Input: An array of characters, W (the word to be analyzed)
Output: An array of integers, T (the KMP table)

Method:

```
pos = 2 ;
cnd = 0 ;
T[0] = -1 ;
T[1] =0 ;

while ( pos < |W| ) do {
    // case I: the substring continues
    if ( W[pos − 1] == W[cnd] ) {
        cnd = cnd + 1 ;
        T[pos] = cnd ;
        pos = pos + 1 ;
    }

    // second case: substring does not continue, but we can fall back
    else if ( cnd > 0 )  {
        cnd = T[cnd]  ;
    }

    // third case: we have run out of candidates. Note cnd = 0
    else {
        T[pos] = 0 ;
        pos = pos + 1 ;
    }
} // end while.
```

Example Where "Fall Back" Occurs:

i	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4
W[i]	A	B	A	B	A	B	X	A	B	A	B	A	B	Y	Y
T[i]	-1	0	0	1	2	3	4	0	1	2	3	4	5	6	0

```
Fallback:  pos =  7,  old value of end = 4,  new value of end = 2
Fallback:  pos =  7,  old value of end = 2,  new value of end = 0
Fallback:  pos = 14,  old value of end = 6,  new value of end = 4
Fallback:  pos = 14,  old value of end = 4,  new value of end = 2
Fallback:  pos = 14,  old value of end = 2,  new value of end = 0
```