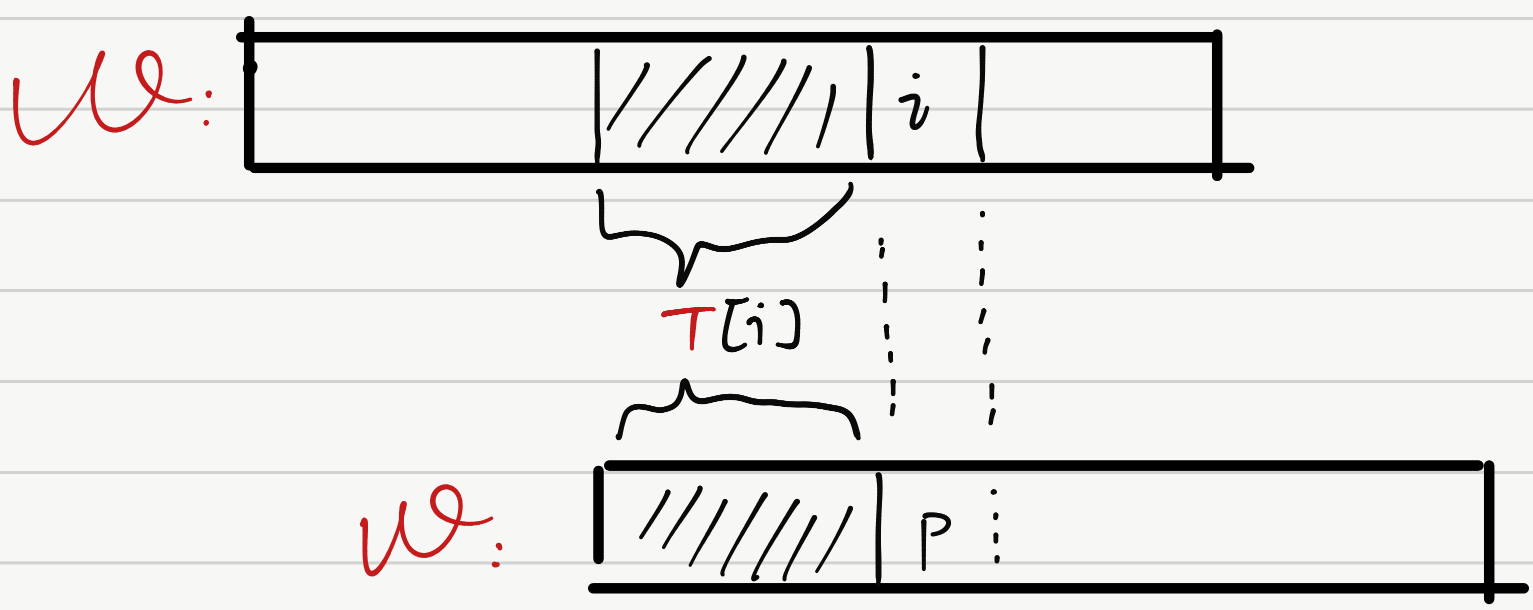
Doc

2021年9月8日 下午11:25

For matching a word — W, the KMP algorithm computes a table T of same length as W.

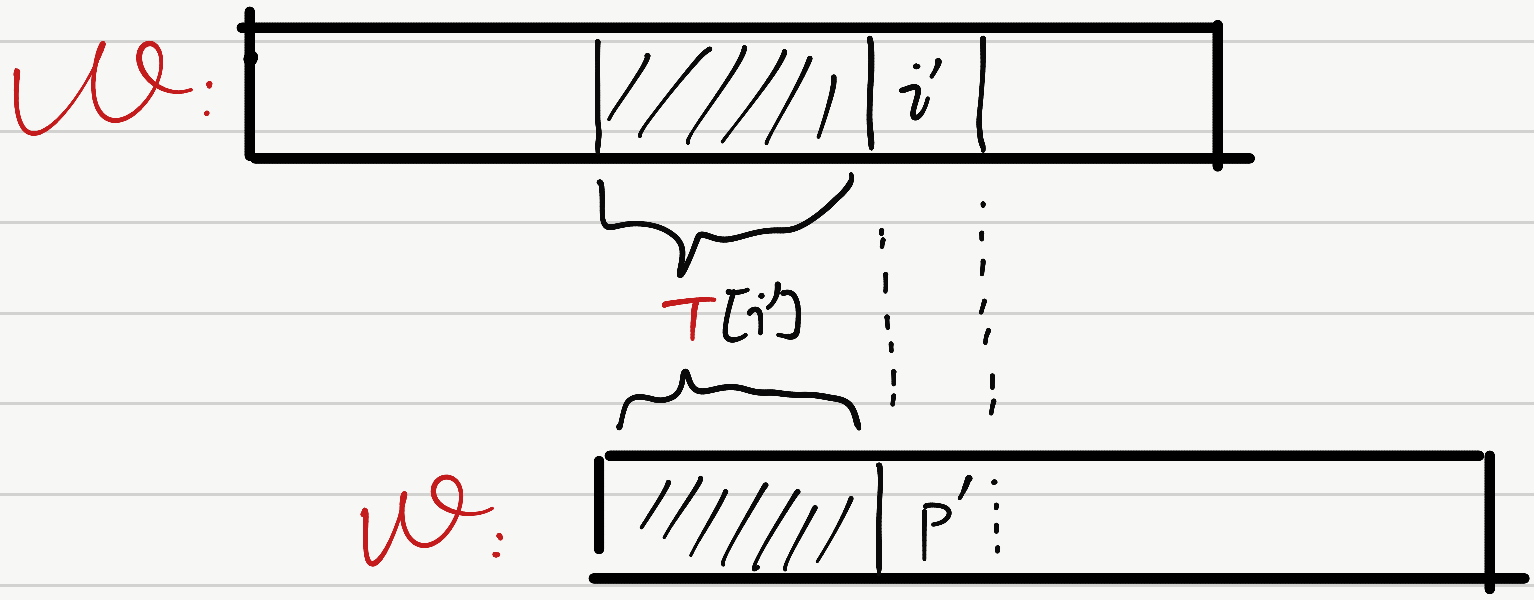
$T[i]$: the length of the longest prefix of W such that it matches the $T[i]$ elements before $W[i]$. I.e.,



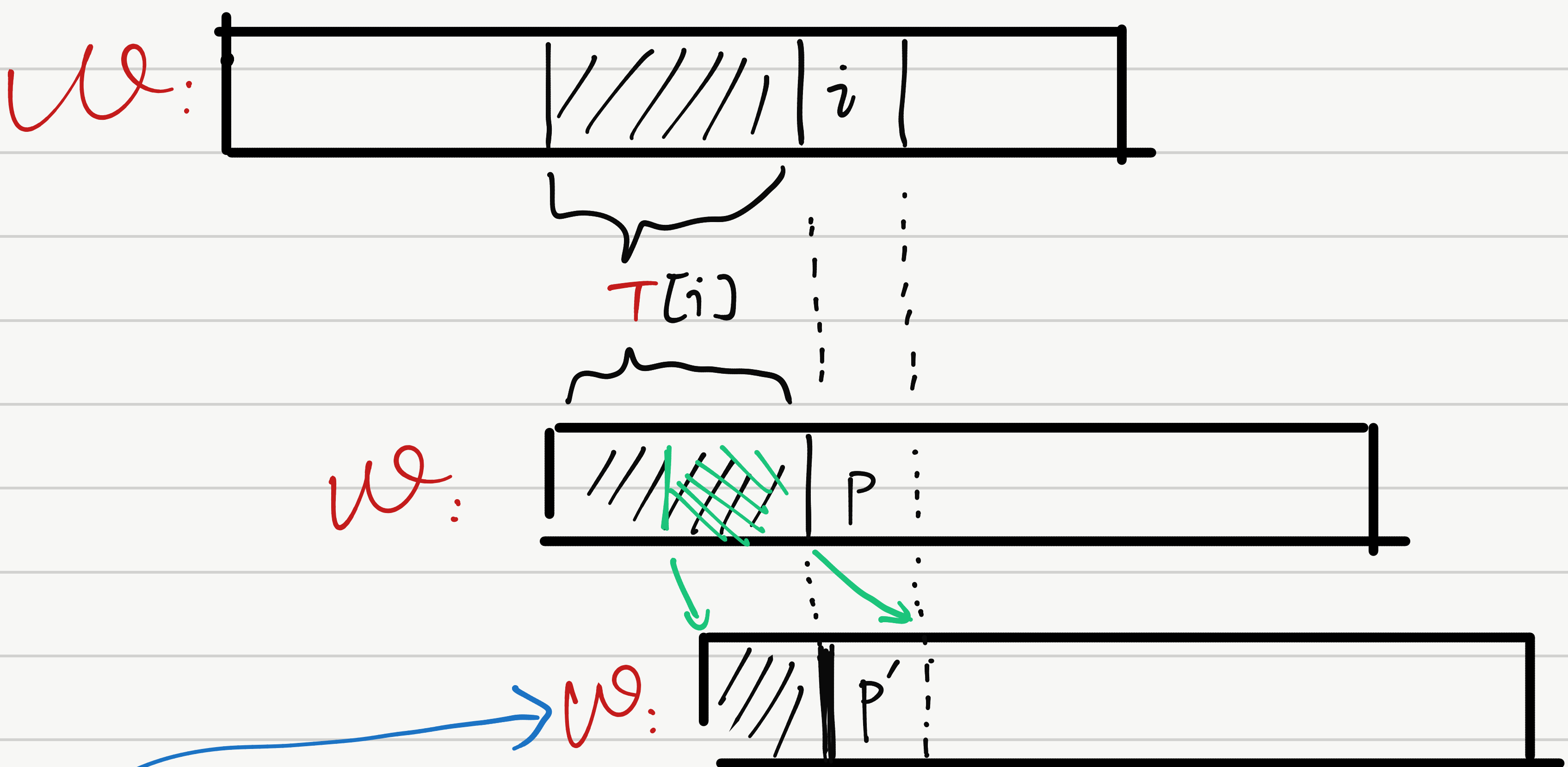
So now imagine the general case of T table computing here:

① $W[i] = W[p]$, then we simply move forward:

$i' = i+1$; $p' = p+1$; $T[i'] = T[i]+1$



② $w[i] \neq w[p]$, then we want push p back to the "closest" place p' such that



the prefix of w ends at p' matches the segment ends at i. Then we can continue the computation

By observing this, we found that $p' = T[p]$ if such p' exists. Recall the definition of T.

This part is the hardest to think of.

To test whethet p' is good so that we can move forward, we just went back to the same situation when we compare $W[i]$ and $W[p]$.

We keep doing this until either we found such p' or no such p' exists. If the latter case, we match $i+1$ with $W[0]$ next.