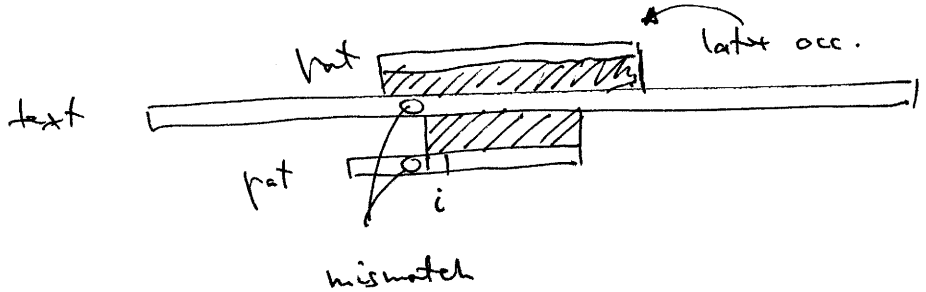
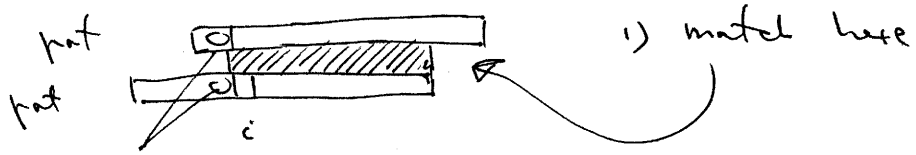
Preprocessing for BM

Recall that after a mismatch, we for the def. of BM_Shift consider a next possible occurrence:



This leads us to consider least possible shift s.t.:



2) mismatch here (unless we are considering shifts of $\geq i-1$ (where there is no char in top copy of pat))

We would like that value

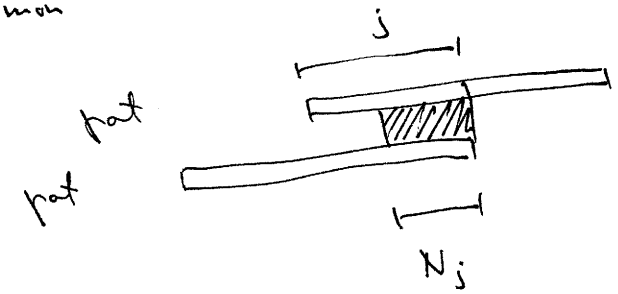
for all $i \geq 2$ (and $i \leq m+1$). [BM_Shift(i) is then that value.]

For $i = 1$, we have an occ., and consider ~~large~~ least possible shift ≥ 1 such that 1) holds [2) will be in "unless" case].

[BM_Shift(1) is then that value]

[Note: i here is one larger than there corresponding value (called i) in textbook BM alg. on page 28 (as it is there position of ~~match~~ mismatch, not last match). Our exposition here corresponds to handout from Gusfield book]

First, define for $j = 1, 2, \ldots, |pat|$ N_j as largest (common) suffix of $pat[1..j]$ and pat:

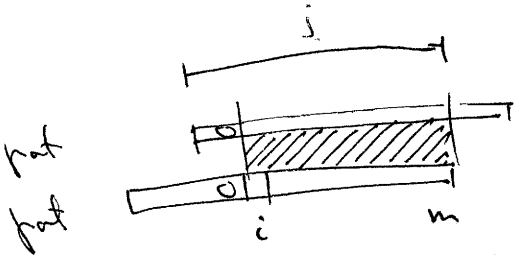


Observe that (by left-right mirror-flip of figure) the N_j values can (be found from) the prefixtable values Z_i for the reverse string (and for $i = j - m + 1$) are equal to.

Hence, the N_j values can be found in $O(|pat|)$ time.

for $i \geq 2$ and $i \leq m+1$

Let $L'(i)$ be largest $j < m$ s.t. $pat[1..j]$ has $pat[i..m]$ as suffix and $j - (m - i + 1) \geq 1$ and $pat[i-1] \neq pat[j - (m - i + 1)]$



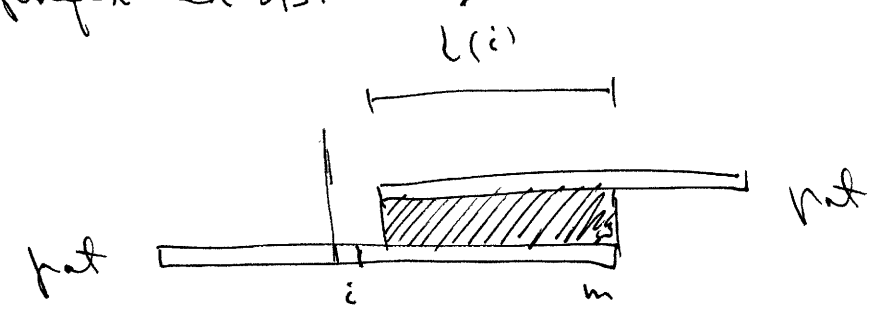
$(m = |pat|)$

Let $L'(i) = 0$ if no such j exists. Let $L(1) = 0$.

I.e., if $L'(i) > 0$, this gives BM_Shift(i) for $i \geq 2$. (as $m - L'(i)$).

(if 2) is not in unless part)

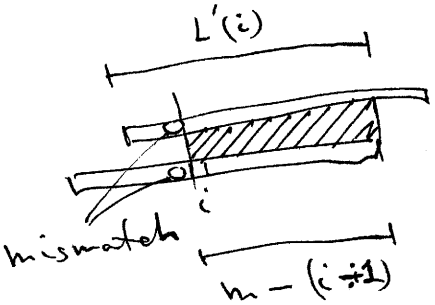
Let $L'(i)$ = length of largest suffix of pat[i..m] being a prefix of pat. Let $L'(i) = 0$ if no such prefix exists.

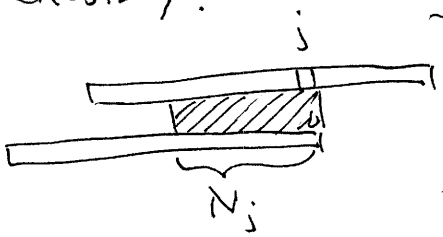


I.e., this gives BM_Shift(i) [as $m - L'(i)$] if $L'(i) = 0$ [if our "shift search" enters the "unless" part of 2)].

(Note: $L'(1) = m$ always. This corresponds to occ., and a shift of ≥ 1 is needed. This shift is $L'(2)$. So we should set $L'(1) = L'(2)$. by above def.)

Observe: $L'(i)$ = largest $j < m$ where $j \geq m - i + 2$ AND $N_j = m - i + 1$ (or $= 0$ if no such j exists).

[Compare 

with 

]

$N_j = m - i + 1$ ⟺ $i = m - N_j + 1$ *) ← (AND $j \geq m - i + 2$)

[Note: We can leave out the "AND" (and hence the "if" in code) if we allow L' to cover 1st "unless" case of *) [Think]]

Given the N_j table (prefix table), the non-zero $L'(i)$ values [those where a j exists] will be those hit by *) when j goes from 1 to $m-1$. For each of these, we should keep the largest j as $L'(i)$.

So the alg.:

for $i = 1$ to $m+1$:
  $L(i) = 0$

for $j = 1$ to $m-1$:
  $i = m - N_j + 1$
  if $j \geq m - i + 2$
    $L(i) = j$

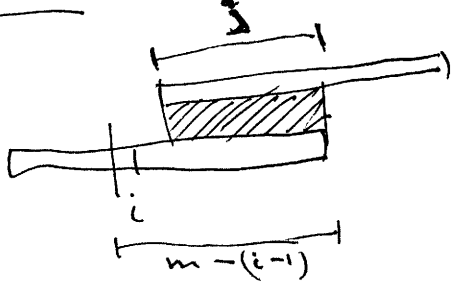
Note: For these j ($\leq m-1$), the possible values of N_j are $0, 1, -, m-1$, so possible values for i are $2, 3, -, m+1$. Which is what we wanted.

is correct. Clearly $O(m)$.

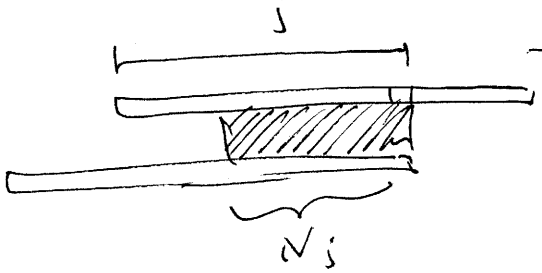
(or zero if no such j exists)

Observe: $L'(i) =$ "largest $j \leq m - i + 1$ s.t. $N_j = j$"

[Compare:



with

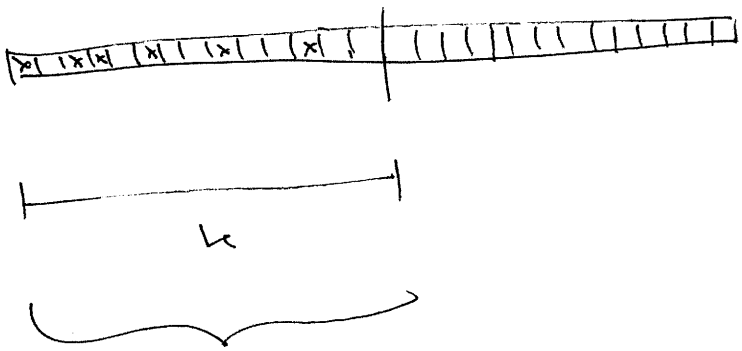


]

Let $k = m - i + 1 \iff i = m - k + 1$

Let $A(j) =$ "$N_j == j$"

Truth-table for $A(j)$:



$$l'(i) = \max_{j \le k} \{ N_j = j \}$$

$k = m - i + 1$

So alg. for $l'(i)$:

```
max = 0
for k = 1 to m-1
    if N_k = k
        max = k
    l'(m-k+1) = max
                (i)
l'(1) = l'(2)
l(m+1) = 0
```

Recall:

$l'(1) = m$ by above def. of l' [see page ③].

So we should set $l'(1) = l'(2)$, as we need a shift ≥ 1.

If $i = m+1$, there are no possible unless-case matches, so set $l(m+1) = 0$

Clearly $O(m)$.

Recall: If $L'(i) > 0$, we should use $L'(i)$.
Else, we should use $l'(i)$.
(by our defs. $L'(1) = 0$ and $L'(1) = l'(2)$, this also covers the case $i = 1$ [i.e., an occurrence].)

In other words, we should use

$$\max\{L'(i), l(i)\}$$

(if $L'(i) > 0$, then $L'(i) \geq l'(i)$, by the way we define our shifts (for a given i, the j's tried for L' are bigger than those for l' (the 'unless' case in 2))).

So finally, create BM_shift by:

> For $i = 1$ to $m+1$.
>
> $$BM\_shift(i) = m - \max\{L'(i), l'(i)\}$$

Total time is $O(m)$.