# KMP Shift Function

Recall $B(k)$ = largest nontrivial border of pat[1..k].

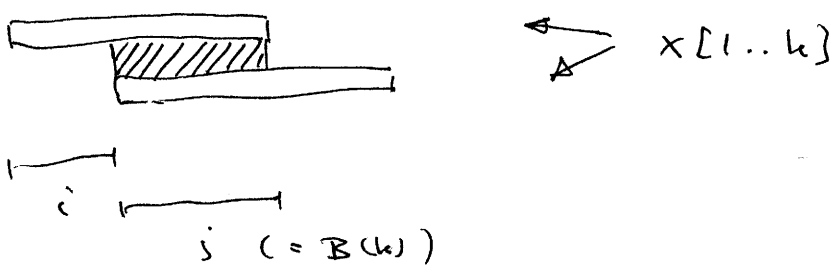
(For brevity, call pat for x below.)

We will compute $B(k)$ for increasing $k$. (In a way that will only reference already computed B values.)

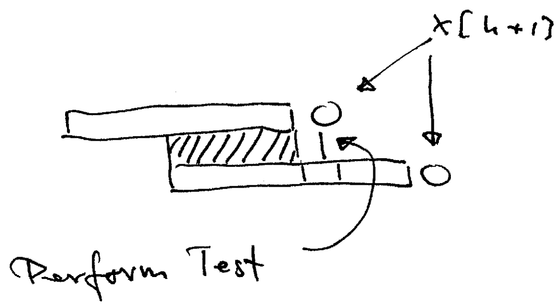
Let $j = B(k)$

$i = k - j \quad (\Leftrightarrow k = i + j)$

So for $B(k)$ we have the picture




Looking at $B(k+1)$, we consider adding $x[k+1]$:




Clearly, $B(k+1) \leq B(k) + 1$ [if border can be enlarged (more than above) after we add $x(k+1)$

to picture, we could have done it before ]. So a positive test means $B(k+1) = B(k)+1$, a negative means $B(k+1) < B(k)+1$.

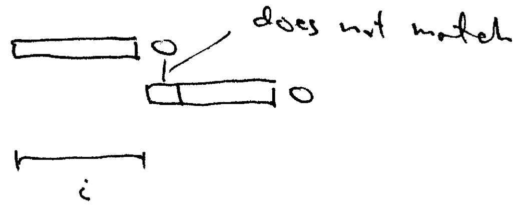
<u>Case A</u> : $x[k+1] = x[j+1]$

We perform the update (advancing $k$)

$$j = j+1$$
$$B(k) = j$$

<u>Case B1</u> : $x[k+1] \neq x[j+1]$ and $j = 0$

I.e:




By $0 \le B(k+1) < B(k)+1 = 0+1 = 1$

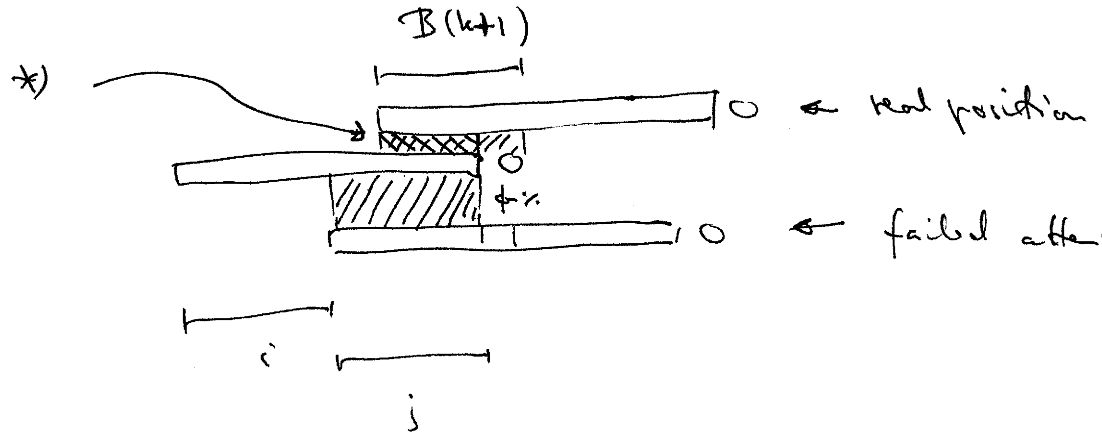
we have $B(k+1) = 0$.

We perform the update (advancing $k$)

$$i = i+1$$
$$B(i+j) = 0$$

<u>Case B2</u> : $x[k+1] \neq x[j+1]$ and $j \geq 1$

We know that "real position" of $B(k+1)$ is strictly to left of current "attempted position" :




Since at real position, *) - part must match,
we can safely move $j - B(j)$ to the right
(this is the first shift $\geq 1$ to right giving match
at *) ) without skipping "real position" of $B(k+1)$.

Additionally, this part *) <u>will</u> match after such
a shift (as $B(j)$ corresponds to an actual border).

So we can start testing at same position.
We therefore do the update

$$i \to i + (j - B(j))$$
$$j \to j - (j - B(j)) = B(j).$$

(This does not advance $k$, and we have not
found $B(k+1)$ yet).

By the convention $B(0) = -1$, we can merge the updates on $i$ and $j$ in cases B1 and B2 into

$$i = i + (j - B(j))$$
$$j = \max\{0, B(j)\}$$

This leads to the following algorithm:

$$\left.\begin{array}{l} B(0) = -1 \\ B(1) = B(2) = \cdots B(|x|) = 0 \\ i = 1 \\ j = 0 \end{array}\right\} \text{initialization}$$

<u>while</u> $(i+j) < |x|$

  <u>if</u> $x[i+j+1] = x[j+1]$

    $j = j+1$

    $B(i+j) = j$

  <u>else</u>

    $i = i + (j - B(j))$

    $j = \max\{0, B(j)\}$

Correctness follows by seeing that the case analysis A, B1, B2 gives the following invariants for alg.:

For $k' \le k (= i+j)$, $B(k')$ is correct

For $k' > k (= i+j)$, $B(k')$ is $0$.

(Recall that $B(1) = 0$ always.).

$\begin{pmatrix} k = i+j \\ \Delta k = \Delta i + \Delta j \end{pmatrix}$

| Case | $\Delta i$ | $\Delta j$ | $\Delta k$ |
|---|---|---|---|
| A | 0 | 1 | 1 |
| B1 | 1 | 0 | 1 |
| B2 | $\geq 1$ | $< 0$ | 0 |

By the column for $\Delta k$, and the loop condition, we see $k \leq |x|$ always.

By $j \geq$ always, we have $i \leq i+j = k \leq |x|$ always.

As initially $i = k = 1$, and as $\Delta k$ or $\Delta i$ increases at least one each iteration (and never decrease), we have at most

$$2 \cdot (|x| - 1) \text{ iterations.}$$

So alg. takes time $\Theta(m)$

$$(m = |x| = |pat|).$$