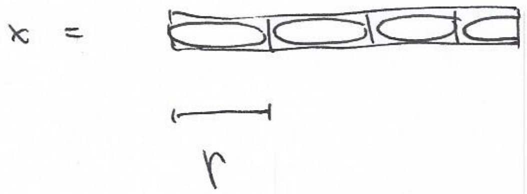
Galil Variant of Boyer-Moore

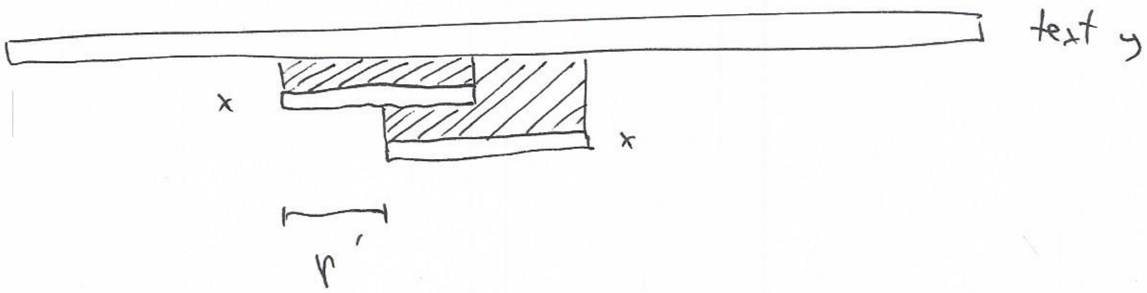
Adds idea to search after an succesful iteration in BM.

Let p be the period of the pattern x [i.e., $p = \mathrm{per}(x)$]

With $\bigcirc$ denoting the prefix of x of length p, we have



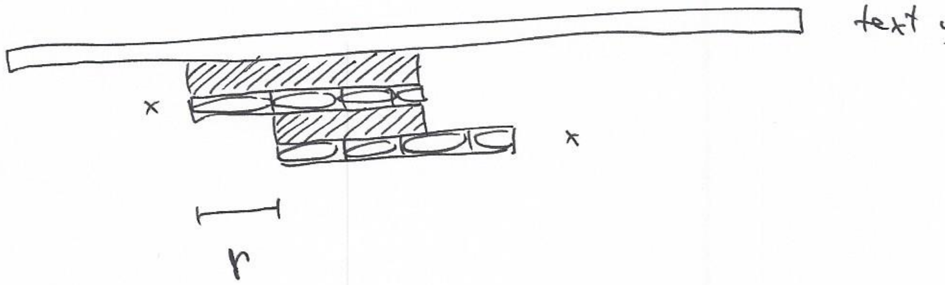
Consider two consecutive matches :



We see that p' is some period of x. As the period p is the shortest of all periods, we have $p \leq p'$.

Hence, it is safe (no occurrences will be skipped) to shift p after a succesful match.

Additionally, we know that after such a shift, the first $m-p$ character will match ($m = |x|$):



Thus, if we after the shift in the backwards scanning of the window reaches down past position $m-p$ in x, we again have a match (and can again shift by p). If not, we have an unsuccesful attempt, and shifts according to the usual BM rules (using good suffix table).

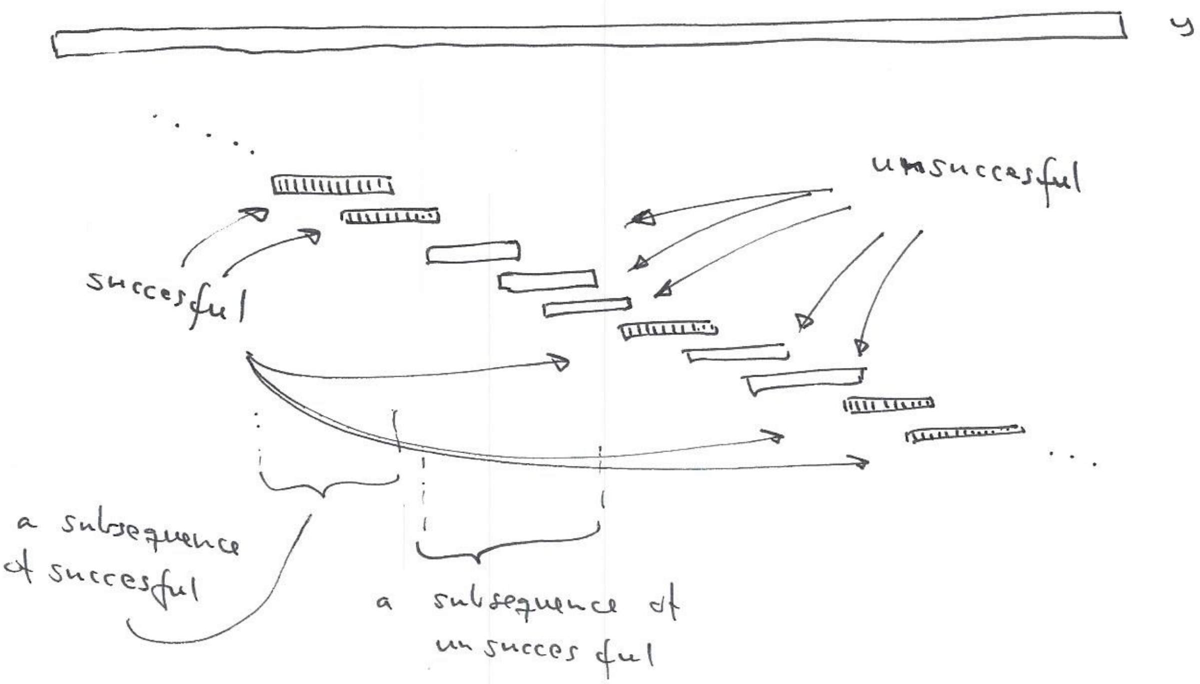
In the implementation, we use a variable l to hold the length of a known matching prefix. It always has one of the two values 0 and $m-p$.

The resulting code (Galil variant of BM) can be seen in the textbook on page 112 (under a different, long name).

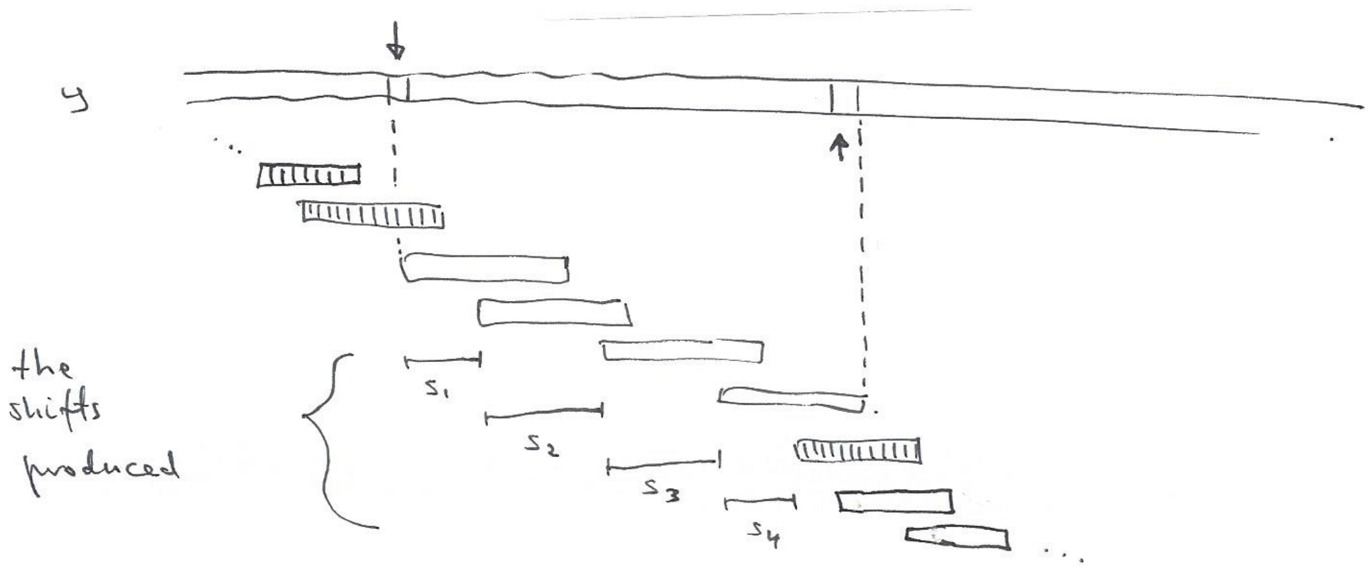
Correctness is clear from discussion above (and safeness of good-suffix shift after unsuccesful attempts).

We now analyze the complexity (for reporting <u>all</u> occurrences).

The entire algorithm is a sequence of attempts (window positions), some succesful, some unsuccesful. In general:



First look at a subsequence of unsuccesful attempts:



The work done in this subsequence is exactly the same as would be done if standard BM would search for x in y truncated to start at the beginning of the first attempt / window position in this sequence. I.e. if y started at the arrow $\downarrow$.

For attempt i in this subsequence of unsuccesful attempts, let t_i be the number of chars of y compared for the first time [during this subsequence] and let s_i be the shift produced. From the observation at top of this page, we can apply the analysis of the standard BM algorithm [Sec. 3.2 in book]. Hence, we know that the work (number of comparisons) at attempt i is bounded by

$$3s_i + t_i$$

[See proof of Thm. 3.7 on page 111, where s_i is called d and t_i is called t].

So the total work during this subsequence is at most

$$*) \qquad \sum_i (3s_i + t_i) = 3\sum_i s_i + \sum_i t_i$$

Clearly (by meaning of t_i) the sum $\sum_i t_i$ is at most equal to the stretch of y lying between the two arrows $\downarrow$ and $\uparrow$.

This can overlap at most m chars with the corresponding stretch for the next subsequence of unsuccesful attempts. So the value $t = \sum_i t_i - m$ can be charged uniquely to chars of y, with no overlap to the same charging for the next (or any other) subsequence of unsuccesful attempts.

Let r be the number of subsequences of succesful attempts. Then there are at most $r+1$ subsequences of unsuccesful attempts. If T is the sum of all t_i's for all these unsuccesful subsequences, we have

$$T - (r+1)\cdot m \leq |y| \quad (= n)$$

Actually, the last unsuccesful sequence does not need the "$-m$" (there is no next subsequence to overlap with), so we get

$$T \leq n + r \cdot m$$

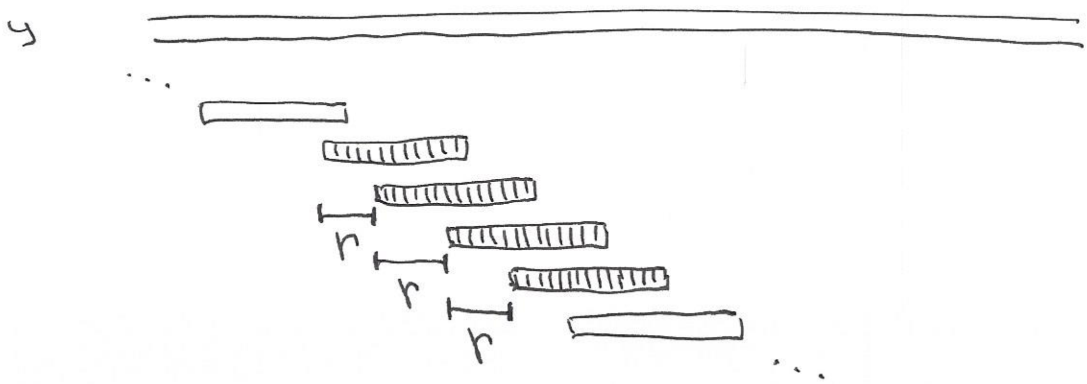
The s_i values for different subsequences (and within each subsequence) can clearly be charged to chars of y without overlap. So if the sum of all these s_i values is denoted S, we have

$$S \leq n .$$

By *) on page 4, the total work of all unsuccessful attempts is at most

$$3 \cdot S + T \leq 4 \cdot n + r \cdot m \qquad (1)$$

Next look at a subsequence of succesful attempts. The first attempt in the sequence will compare m chars. The next will be shifted p, and will compare p chars (see page 2). The latter will also apply to all remaining attempts in the subsequence



Thus, except for the first m comparisons, the work of the subsequence can be charged to chars of y in a way not overlapping the charging from other subsequences of succesful attempts.

Thus, with r such subsequences, their total work is bounded by

$$n + r \cdot m \qquad (2)$$

We now bound r.

<u>Case 1</u>: $p \geq m/2$. Since p is a lower bound on the distance between succesful attempts (see page 1), we know that if there are $\tilde{r}$ succesful attempts in total, we have

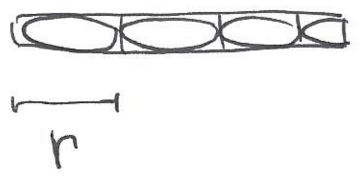
$$m + (\tilde{r} - 1)p \leq n$$
$$\Updownarrow$$
$$\tilde{r} - 1 \leq \frac{n-m}{p} \leq \frac{n-m}{m/2} = 2\frac{n}{m} - 2$$
$$\Updownarrow$$
$$\tilde{r} \leq 2\frac{n}{m} - 1$$

Clearly, $r \leq \tilde{r}$, so we know $r \leq 2\frac{n}{m}$

Summing (1) and (2) and plugging in $r \le 2\frac{n}{m}$ we get a bound on the total work [more precisely, total number of comparisons] of

$$5 \cdot n + 2 \cdot r \cdot m \le 9 \cdot n$$

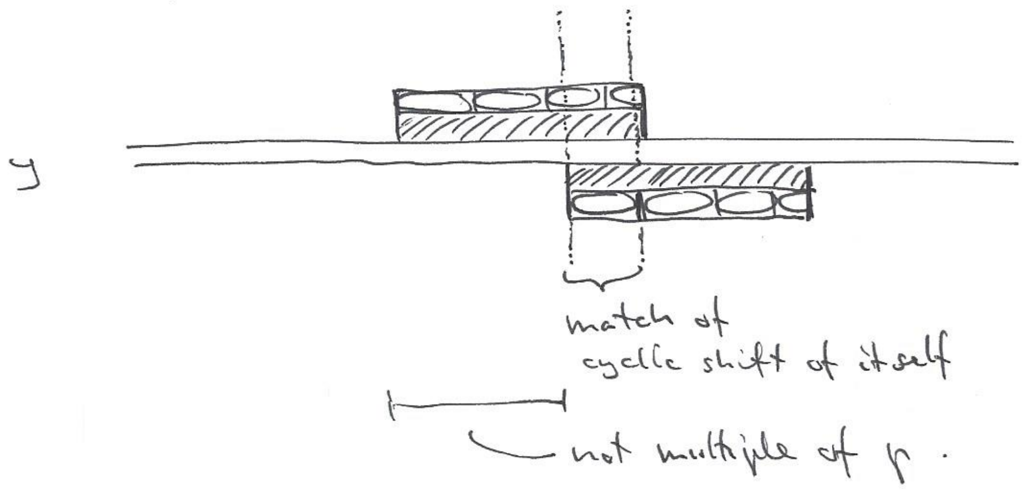
Case 2 : $p \le m/2$. Recall (page ①) the shape of x :

$x =$ 

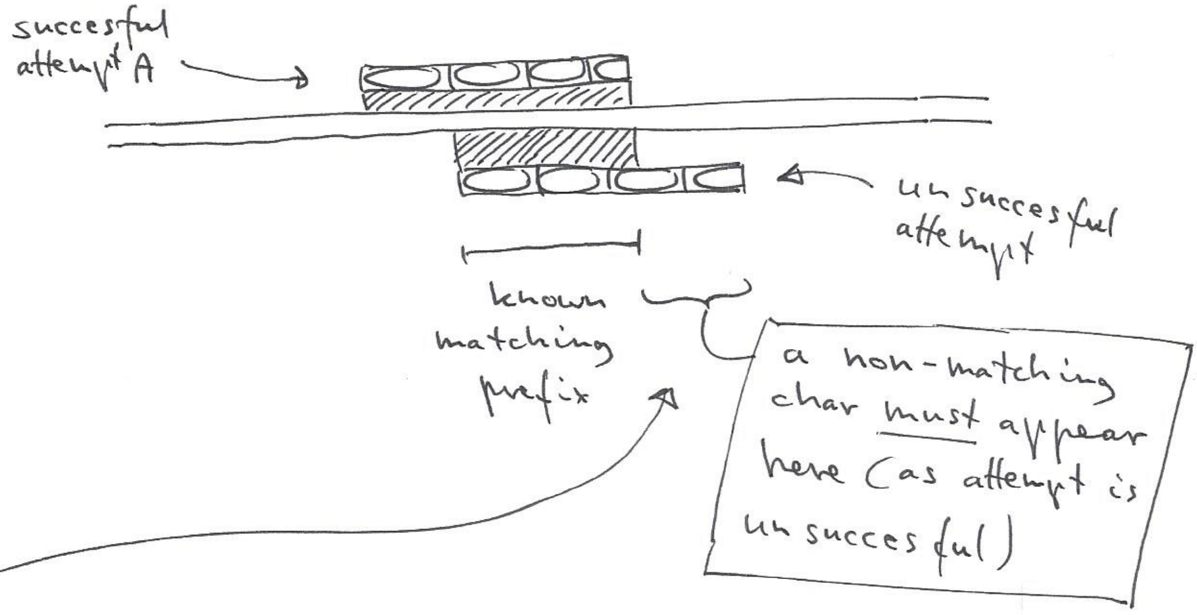
The string  (the prefix of x of length p) is not a power of any other string (else, p would not be the shortest of all periods), i.e., it is primitive.

Hence, we know it is not equal to a (proper) cyclic shift of itself [see teachers note on primitive strings].

This implies that *if* two occurrences overlap by p or more characters, they *must* be "aligned", i.e. differ in position by a multiple of p – else $\boxed{}$ would match a cyclic shift of itself:



Look at last attempt A in a succesful subsequence (i.e. the next attempt, which is after a shift of p, is unsuccesful):



Any window position aligned (i.e., differing by a multiple of p) with A (and overlapping A) must also have a character not matching with y in this area, as the same chars of x will appear there. So the next occurrence after A <u>cannot</u> overlap A by p chars or more (cf. bottom of page 7).

We have proven that between the last occurrence in a subsequence of succesful attempts and the first occurrence of the next such subsequence, there is a shift of more than $m - p$, which in Case **2** is at least $m - m/2 = m/2$.

Similar to page 7 [but looking at subsequences of positive attempts rather than individual occurrences] we get

$$m + (r-1) \cdot p \leq n$$

$$\Updownarrow$$

$$r \leq 2 \cdot \frac{n}{m} - 1$$

As on page 8 this gives a bound on the total work of $9 \cdot n$.

We have proven: the galil variant of BM uses $O(n)$ time to find <u>all</u> occurrences.

Finally, we note that the period p is the first entry in the good-suffix table for BM (as well as the last entry of the borders table for KMP).

Hence, preprocessing is still $O(m)$ for the Galil version of BM.