← K divide a N.

Entrada: $S[0..N)$, $N>0$, y $K>0$ tal que $K|N$

Salida: Mínima cantidad de pedazos requeridos para codificar $S[0..N)$ reorganizando individualmente cada uno de los N div K bloques de tamaño K de $S[0..N)$.

Sea $M = N \operatorname{div} K$. El arreglo $T[0..M)$ es tal que, para $0 \leq m < M$:

$T[m] \subseteq \{'a', 'b', \ldots, 'z'\}$ es el conjunto de letras que hacen parte del m-ésimo bloque de $S[0..M)$; es decir $T[m]$ son las letras en $S[mK..(m+1)K)$.

Dado que hay 26 letras minúsculas, vamos a suponer que cada $T[m] \subseteq \{0, \ldots, 26\}$, en donde el número 26 identifica una letra que no aparece en $S[0..N)$.

Con base en esto, se propone la siguiente función para $0 \leq m \leq M$ y $0 \leq c \leq 26$:

$\phi(m, c)$: "Mínima cantidad de pedazos requeridos para codificar $T[m..M)$ en donde el último bloque (antes de m) termina con el caracter c".

Objetivo:

$\phi(0, 26)$ Se busca la menor cantidad suponiendo que en el bloque 0 se inicia con una letra nueva (26 no está en $S[0..N)$)

Recurrencia: Para $0 \leq m \leq M$ y $0 \leq c \leq 26$:

$$
\phi(m, c) = \begin{cases}
0 & , m = M \\
|T[m]| + (\downarrow i \mid i \in T[m] : \phi(m+1, i)) & ①, m \neq M \wedge c \notin T[m] \\
\phi(m+1, c) & ②, m \neq M \wedge \{c\} = T[m] \\
\left[(|T[m]| - 1) + (\downarrow i \mid i \in (T[m] \setminus \{c\}) : \phi(m+1, i)) \right] & ③, m \neq M \wedge \{c\} \subsetneq T[m] \\
\quad \downarrow (|T[m]| + \phi(m+1, c)) &
\end{cases}
$$

Sobre algunos casos de la recurrencia:

① c no está en $T[m]$, luego se requieren $|T[m]|$ pedazos (uno por caracter en $T[m]$) y se prefiere como caracter para terminar aquel que está en $T[m]$ que minimice la codificación para $T[m+1..M)$

② c es el único caracter en $T[m]$, luego no se deben crear nuevos pedazos para codificar $T[m]$ y se recurre usando c como caracter final de $T[m]$.

③ c es uno de los caracteres en $T[m]$; hay dos casos:
- iniciar la codificación de $T[m]$ con c y preferir el 'mejor' de los demás caracteres para terminar la codificación de $T[m]$
- terminar la codificación de $T[m]$ con c, creando un total de $|T[m]|$ pedazos para codificar $T[m]$.