Subtarea 1

- $a_i = i$

En esta subtarea, $a_r - a_l = r - l$, por lo que siempre es posible ir de i a cualquier $j \in [i+1, i+k]$ en 1 salto. Por tanto, la respuesta para una pregunta (l, r) es $\lceil \frac{r-l}{k} \rceil$.

Subtarea 2

- $a_i - a_{i-1} = c$, para alguna constassnte c.

La idea es similar. En esta subtarea, $a_r - a_l = (r - l) \cdot c$.

Entonces, si queremos ir de i a j en un salto, se tiene que cumplir que

$(j - i) \cdot c \leq k \Leftrightarrow (j - i) \leq \lfloor \frac{k}{c} \rfloor$. Por tanto, podemos dar saltos de longitud $\lfloor \frac{k}{c} \rfloor$.

Entonces, la respuesta para una pregunta (l, r) es $\left\lceil \frac{r-l}{\lfloor \frac{k}{c} \rfloor} \right\rceil$.

Subtarea 3

- $n \cdot q \leq 10^6$.

Podemos resolver cada pregunta en tiempo lineal. Podemos usar dos punteros para dar los saltos óptimos, manteniendo para cada l, el mayor r tal que $a_r - a_l \leq k$.

Complejidad temporal: $\mathcal{O}(n \cdot q)$.

Subtarea 4

Subtarea 5

Digamos que $f(l)$ es el mayor $r \geq l$ tal que $f(r) - f(l) \leq k$. Si no existe tal r, digamos que $f(r) = -1$.

$f()$ se puede computar en tiempo lineal usando dos punteros, de derecha a izquierda.

Podemos considerar esto como un grafo dirigido, donde cada nodo tiene grado de salida 1. De hecho, si consideramos los $x : f(x) = -1$ como raíces, este grafo es realmente un bosque con las aristas orientadas desde las hojas hacia la raíz.

Podemos usar la técnica **binary lifting**, que consiste en mantener para cada nodo, la lista de ancestros a distancia $2^0, 2^1, 2^2, 2^3, \ldots$.

Si tenemos una pregunta (l, r), usando esta lista de ancestros, podemos vorazmente buscar el primer ancestro de l que sea mayor o igual que r, y la respuesta sería la diferencia entre las profundidades de l y el ancestro encontrado en el grafo.