Subtarea 1

- $a_i = 1$

Entonces, $f(l, r) = 1$ por lo que la respuesta es $\frac{n(n+1)}{2}$

Subtarea 2

- $a_i \in \{1, 2\}; n \leq 100$

La solución de la subtarea 3 sirve para esta subtarea.

Subtarea 3

- $n \leq 100$.

Es suficiente iterar por todos los rangos $[l, r]$ y computar por cada uno, la cantidad de elementos distintos en tiempo lineal. Para hacerlo, llevamos un arreglo de marcados, tal que `mark[x] = true` significa que x aparece en el rango analizado. Luego, la respuesta es la cantidad valores `true` en `mark[]`.

Complejidad: $\mathcal{O}(n^3)$.

Subtarea 4

De la subtarea anterior, estamos computando varias cosas múltiples veces. Observemos que entre el rango $[l, r]$ y el rango $[l, r+1]$ no hay mucha diferencia en la información que se lleva. En lugar de reiniciar `mark[]` cada vez e iterar por $[l, r+1]$ modificando `mark[]`, podemos simplemente fijar l, luego iterar por r en orden creciente, y actualizando solo $mark[a_r]$ en cada momento. Cada vez que `mark[x]` cambia de `false` a `true`, le añadimos 1 a la respuesta.

Complejidad: $\mathcal{O}(n^2)$

Subtarea 5

Digamos que $nxt(i)$ es el primer $j > i : a_j = a_i$, y si no existe tal j, digamos que es $n+1$.

Podemos considerar $f(l, r)$ como la cantidad de $i \in [l, r] : nxt(i) > r$.

Entonces, la "contribución" de cada i a la respuesta es la cantidad de rangos $[l, r] : i \in [l, r]$ y $nxt(i) > r$.

Tenemos que $1 \leq l \leq i$, y que $i \leq r < nxt(i)$. Por tanto, la contribución de i a la respuesta es $(nxt(i) - i) \cdot i$.

La solución entonces se puede expresar como $\sum_{i=1}^{n} (nxt(i) - i) \cdot i$. Esto tome tiempo lineal.

El cuello de botella de la solución está en computar $nxt(i)$. Podemos hacerlo en tiempo cuadrático simplemente iterando por $j > i$ hasta que encontremos un $j\colon a_j = a_i$, lo cual es suficiente para aceptar esta subtarea.

Para subtarea, simplemente computamos $nxt()$ en tiempo lineal en total. Cómo, podemos ir de derecha a izquierda, actualizando la posición de cada valor.