**H - Frequent Prime Ranges**

**1. Reducción del Problema  
Descripción en palabras:** Se define un rango [L..H] como K-frecuente si contiene al menos K números primos. Dado un entero N y un valor K, se debe contar cuántos subrangos del intervalo [2..N] son K-frecuentes.

**Descripción en lenguaje matemático**: Contar cuántos subrangos [L..H]⊆[2..N] cumplen que π(H)−π(L−1)≥K, donde π(x) es la función que cuenta los primos ≤ x.

**2. Reducción de la Solución  
Descripción en palabras**: Se precalculan los primos hasta N y sus acumulados, y luego se recorren todos los subrangos posibles en [2..N], verificando si contienen al menos K primos.

**Descripción en lenguaje matemático**: Sea P(i)la cantidad de primos en [2..i], entonces un subrango [L..H] es válido si P(H)−P(L−1)≥K; se cuentan todos los pares (L,H)que cumplen esta condición.

**3. Código Realizado y Análisis**

* Enlace(s) Código: <https://vjudge.net/solution/61083793/Y8Oys7pCzh3ilcxtu3N8>

**4. Casos de Prueba**

1. **Entrada**: 2 1  
    **Salida**: 1

**Justificación**: El único subrango posible es [2..2], y contiene un número primo (2), por lo tanto cumple que tiene al menos 1 primo.

1. **Entrada**: 5 2  
    **Salida**: 4

**Justificación**: Los subrangos que tienen al menos 2 primos en [2..5] son: [2..3], [2..4], [2..5], [3..5].

1. **Entrada**: 9 3  
    **Salida esperada**: 8

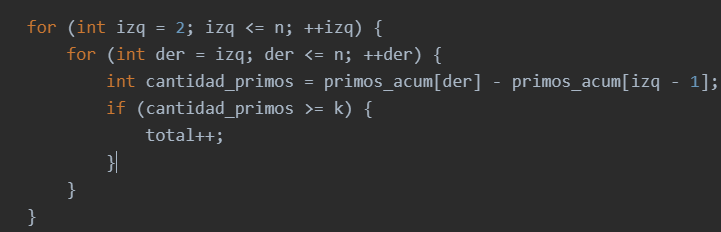
**Justificación**: En el rango [2..9] hay varios subrangos que contienen al menos 3 números primos. Se cuentan exactamente 8 subrangos que cumplen esta condición.

**5. Iteración en Caso de Solución Incorrecta (o explicación Solución Correcta)**

Para este problema lo intente en dos lenguajes diferentes(python y c++) así que mandaré los dos códigos incorrectos.

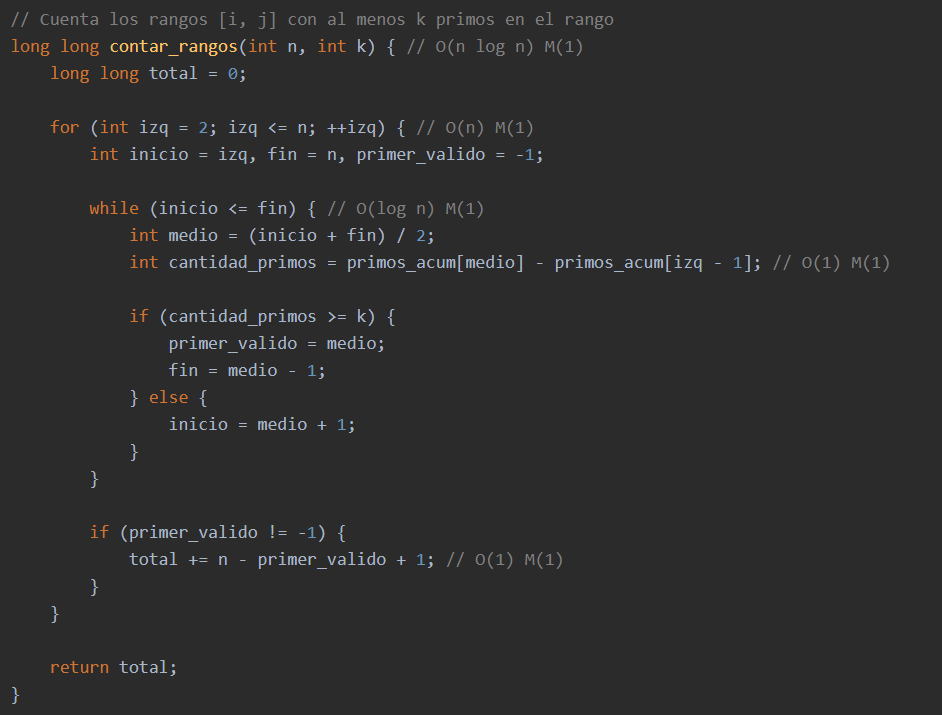
**Códigos incorrecto:**

1. [**https://vjudge.net/solution/61031784/dfkSCBzBjNiYi1Xh623G**](https://vjudge.net/solution/61031784/dfkSCBzBjNiYi1Xh623G)
2. [**https://vjudge.net/solution/61033351/5cHRJEicLDCGFn2Yvxg2**](https://vjudge.net/solution/61033351/5cHRJEicLDCGFn2Yvxg2)

* **Descripción del error**: Primero lo intente en python lo que hacía antes era primero comprobar si el número es primo, y con ello generando dos funciones más la primera que genera una lista hasta el límite y la segunda cuenta los sub rangos pero, pero la manera que lo hice esas dos funciones tenía como complejidad cuadrática y se me pasaba del tiempo luego lo intente c++ con la misma estrategia aunque puse un N máximo pero igual se pasaba del tiempo.
* **Proceso de depuración**: La versión de c++ era qué más eficiente era así que trabaje sobre ella unos días después y error estaba aquí  
  

Este doble for recorre todos los subrangos posibles entre 2 y n. La cantidad de iteraciones es del orden de (n^2), lo cual es demasiado lento cuando n es grande (por ejemplo, 100 000).

y la cambie por esta:



Así que en lugar de probar todos los valores de der uno por uno, este código busca el primer der válido osea tuve que búsqueda binaria, así se reduce el tiempo de cada iteración a O(logn), haciendo el total mucho más eficiente.

* Solución corregida:

<https://vjudge.net/solution/61033361/8k8YCk6xzRKPNkihs4ck>

**6. Preguntas de Aprendizaje**

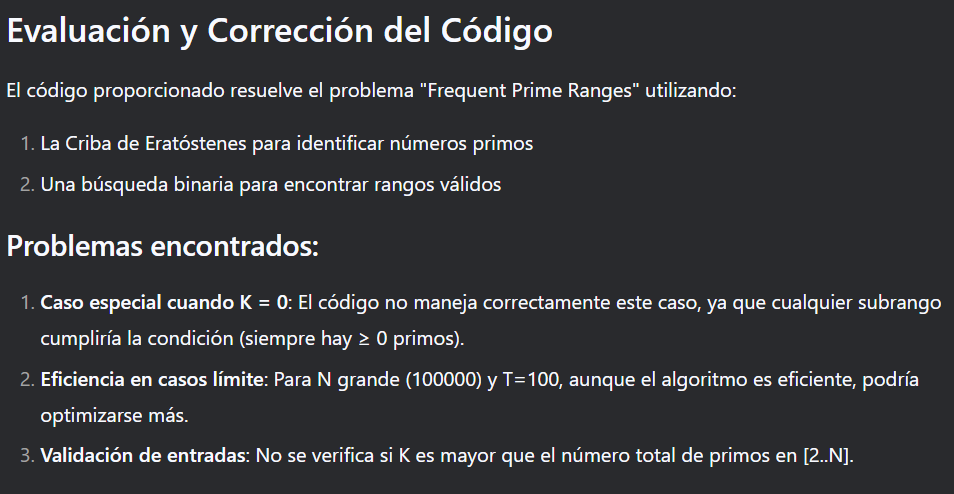
* Temas aplicados:
* Búsqueda binaria
* Criba de Eratóstenes
* Arreglo de acumulados
* Optimización de conteo de subrangos
* Dificultad de la implementación: Media, (hubiera sido fácil si se ocurrido antes lo de la búsqueda binaria)
* Recursos utilizados: Los comentarios de esta pagina <https://www.spoj.com/problems/FRQPRIME/en/> mas videos de youtube

**7. Feedback LLM**

* Envío código LLM: <https://vjudge.net/solution/61083854/pSQXTgAdu59jvVA6MTSa>
* **Comparación de su códigos**: Es baste similar solo que tuvo el mismo error que yo al principio quizás si le dijera que usará búsqueda binaria quizas ahi si lo logré hacerlo más eficiente así que por eso el mio seria mejor

* **Feedback de LLM**: Pida al LLM que evalúe y corrija su código:

Esto fue lo que me dijo



Y si y no estoy de acuerdo osea el caso K=0 si porque nunca tome en cuenta esa caso(porque para el ejercicio no afectaba), y la Validación de entrada no porque no era necesario para términos de este ejercicio, y lo que se pueda optimizarse mas ahi no sabria si esta deacuerdo o no(porque yo no sabría cómo optimizarlo más)