AYUDANDO AL ELECTRICISTA

AUTOR: LAUTARO LASORSA

Solución Esperada

La solución presenta 3 dificultades importantes :

- Modelar correctamente el problema (visualizarlo)
- Identificar qué es lo que está pidiendo y con que cuenta puede lograrse.
- Encontrar una forma eficiente de realizar dichas cuentas.

Respecto del primer item, puede verse de la siguiente manera (aunque no es necesario para resolver el problema) :

"Dado un **Trie**, cuantos conjuntos de X palabras diferentes existen que formen dicho **Trie**"

Lo importante a la hora de modelarlo es tomar la noción de que los tubos y rosetas forman un grafo, un árbol con raíz específicamente :

- Las rosetas son los nodos.
- La roseta 0 es la raiz del arbol. (que en el **Trie** representa a la palabra vacía)
- Los tubos son las aristas del árbol.

Además, los cables son caminos que unen la raíz con alguno de los nodos (la roseta donde termina el cable), y los nodos solo existen si aparecen en al menos uno de los caminos. De lo anterior, se desprende que para cada hoja del árbol debe haber un camino que llegue hasta ella (y, además, con eso será suficiente para que existan todos los nodos).

Las observaciones anteriores se evalúan en la subtarea por 10 puntos donde todas las rosetas están conectadas a la roseta 0. (es decir, solo hay hojas y la raíz). La modelización como árbol jerarquizado también es necesaria para la subtarea por 10 puntos donde N <= 20.

Para las siguientes subtareas será también necesario notar que puedo definir un cable únicamente por la roseta en la que termina. Es decir, defino el camino únicamente por su extremo que es distinto de la raíz.

La siguiente dificultad es identificar correctamente la pregunta y cómo calcular su respuesta.

Deben notar que, como se dijo antes, debe haber un camino que una a cada hoja con la raiz. Por lo tanto :

- Defino a la Cantidad de Hojas del arbol como **CH**.
- A su vez, veo que tengo "nodos interiores", es decir, aquellos que no son ni hoja ni la raiz. Defino la Cantidad de nodos Interiores como **CI** = **M** - 1 - **CH**.
- Observo también que como destine un cable a cada hoja, no me quedan **X** cables. Defino la cantidad de cables que me quedan como Cables Libres, **CL** = **X** - **CH**.

Notar que si X >= M o si X < CH, es imposible armar la red.

Como con abarcar todas las hojas se garantiza la existencia de todos los nodos, cualquier forma de distribuir los cables libres entre los nodos interiores, mientras no haya 2 cables que terminen en el mismo nodo, será válida.

Por lo tanto, se nos está preguntando cuántas formas hay de tomar **CL** elementos de un conjunto de **CI** elementos. Esta observación se evalua en la subtarea por 20 puntos donde N <= 10.000. (ver **Soluciones Parciales**)

Aquí es donde se afronta el tercer problema, el cómo implementar esto de forma eficiente.

La operación es (**CI**!)/(**CL**! * (**CI**-**CL**)!). O también,
(**CI** * (**CI**-1) * (**CI**-2) * ...(**CI**-**CL**+1)) / (**CL**!).

Además, debe observarse que :

- Si se desean factorizar todos los números de 1 a **CI**, puede realizarse en O(CI*log(CI)) utilizando una idea similar a la Criba de Eratóstenes. La idea es tener un vector de vectores **fact**, donde **fact[i]** guarde la factorización de i.

  Recorro todos los números de 2 a **CI** en orden. Si al llegar **fact[i]** se encuentra vacío quiere decir que i es primo. Entonces debo recorrer todos los múltiplos de **i**, y agregarles

a **i** como factor primo. Debo ver qué exponente tiene **i** en esa factorización. Eso puede hacerse fácilmente con `while(k%i==0)k/=i,exp++;`
(asumiendo que **k** es el número y **exp** indica el exponente de i en su factorización)

- Además, recordar que multiplicar 2 números es sumar los exponentes de sus factores primos. Y a su vez, dividirlos es restarle los exponentes del divisor a los del dividendo.

Teniendo en cuenta las 2 cosas anteriores, lo que hay que hacer es :

- Definir un vector cantf que tendrá la factorización del resultado de la operación. **cantf**[i] contiene el exponente de i en la factorización de la respuesta.
- Recorro todos los números entre CI y (**CI**+1-**CL**), y para sus factores primos, sumó sus exponentes en **cantf**. Es decir, si por ejemplo j es un factor primo de k ((**CI**+1-**CL**)<=**k**<=**CI**), y aparece con exponente 2 en su factorización, hago **cantf[j]**+=2.
- Análogamente, recorro los números entre 1 y **CL**, y resto los exponente de sus factores primos de **cantf**.
- Ahora cantf ya tiene la factorización del resultado. Entonces, la respuesta será $(1^{cantf[1]} * 2^{cantf[2]} * 3^{cantf[3]} * \ldots * CL^{cantf[CL]})$%MOD

La complejidad de esta solución es O(**CI***log(**CI**)).

Notar que como el MOD a usar depende del caso de prueba y puede no ser primo, no es posible buscar soluciones alternativas con inverso modular.

Soluciones Parciales

Con la observación de que lo que se está pidiendo es la cantidad de formas de tomar **CL** elementos de un conjunto de **CI** elementos, se puede hacer una solución en O(CI^2) que pasa la subtarea de **N** <= 5.000, valuada en 20 puntos.

Esto puede hacerse utilizando el Triángulo de Pascal (https://es.wikipedia.org/wiki/Tri%C3%A1ngulo_de_Pascal), sabiendo que en la posición **Triangulo[i][j]** se encuentra la cantidad de formas de tomar j elementos de un conjunto de i elementos. Implementativamente puede tomarse una matriz de (**CI**+1) x (**CI**+1), y poner 1 en todos los elementos de la diagonal principal y de la primera columna. Luego, para la fila **i**, para todo **j** (0<**j**<**i**), **Triángulo[i][j]** = **Triángulo[i-1][j-1]** + **Triángulo[i-1][j]**.

En la subtarea donde MOD = 2, valuada en 10 puntos, la respuesta es unicamente si la cantidad de formas es divisible por 2 (se devuelve 0 si lo es, 1 si no). Para eso, hay que comparar si en el dividendo el factor primo 2 tiene un exponente mayor (es divisible) o igual (no lo es) al que tiene en el divisor. Como para cada número solamente hay que ver el exponente del 2 en su factorización, es O(**CI** * log(**CI**)).

A su vez, con la observación de que se puede buscar la factorización de la solución y luego hacer el producto de los factores puede hacerse una solución en O($\mathbf{CI}^{3/2}$), que también se puede decir O(**CI** * raizCuadrada(**CI**)). La diferencia entre esta solución y la solución completa es que en este caso no se precalcula la factorización de cada número, sino que se lo factoriza en el momento. Como factorizar un número k cuesta raizCuadrada(k), el total queda CI * raizCuadrada(CI). Esto pasa la subtarea N <= 50.000, valuada en 20 puntos.