Semana 11 Python

L11A_16-1

Tiempo límite de ejecución: 6 segundos

Problema

La sucesión de fibonacci es una sucesión matemática muy reconocida por su elegante formulación recursiva y su aparición en diversos escenarios matemáticos a veces inesperados. Para este problema consideraremos una versión extendida que llamaremos **k-bonacci**:

$$f(n) = \sum_{i=1}^{k} f(n-k)$$

Naturalmente, se tienen k casos base: $f(0) = a_0, f(1) = a_{1,}, \ldots . f(k-1) = a_{k-1}$ etc.

Para caracterizar una determinada sucesión de **k-bonacci** se usará un conjunto de casos base ordenados. Por ejemplo la lista [1, 1] representa la sucesión clásica de fibonacci con k = 2 (Porque hay 2 casos base): 1, 1, 2, 3, 5, 8, 13, 21, 34 ….

Si tenemos la lista [1, 2, 3, 1], k = 4 la sucesión sería: 1, 2, 3, 1, 7, 13, 24, 45, 89, 171, 239… etc.

n	0	1	2	3	4	5	6	7	8	9
f(n)	1	2	3	1	7	13	24	45	89	171

Por ejemplo f(6) = f(2) + f(3) + f(4) + f(5) = 3 + 1 + 7 + 13 = 24

Calcular la sucesión de **k-bonacci** es demasiado fácil, por lo que su tarea inicial era calcularla *eficientemente*. Sin embargo calcularla eficientemente es bastante fácil… Por lo que su tarea es encontrar el n más *pequeño* tal que $f(n) >= 10^{100}$

Entrada

La primer línea del caso de prueba es un número **T** que representa la cantidad de casos de prueba. Cada caso contiene una única línea: los k casos base.

Restricciones / Consideraciones

$2 <= k, a_i <= 1000$

Salida

Por cada caso de prueba se debe imprimir el mínimo n tal que $f(n) >= 10^{100}$.

Ejemplo de Entrada	Ejemplo de Salida
5 1 1 1 2 3 999 50 4 100 100 100 100 100 100000 88888 1000000000000000000	480 378 371 337 306

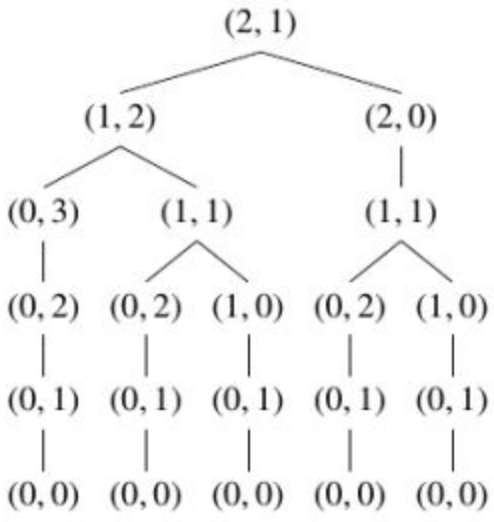
L11B_16-1

Tiempo límite de ejecución: 2 segundos

Problema

La abuela de Pedrito le ha enviado a comprar una par de pastillas para la memoria. Estas pastillas vienen en dos tamaños: grandes y pequeñas. La dosis en cada píldora grande es equivalente a la de dos pequeñas. Luego de volver a casa, pedrito observa que la abuela toma una pastilla al azar todos los días: si se trata de una pastilla pequeña, ella se la toma; de otra manera, se divide y toma la mitad. Después de esto la mitad restante es considerada una pastilla pequeña. Dada una botella con l pastillas grandes y s pastillas pequeñas, decimos que el par (l, s) es la configuración de la botella. Pedrito está interesado en el árbol de las píldoras asociado con la configuración de la botella (l, s), en el que a la izquierda y derecha se representa la ramificación de una píldora grande o pequeña respectivamente.

Formalmente el árbol binario marcado con raíz (l, s) en el que un nodo (u, v) tiene un hijo izquierdo (u-1, v + 1) si u> 0 y un hijo derecho (U, v - 1) si v> 0.
Por ejemplo, el árbol de píldoras asociado a la configuración de la botella (2, 1) (2 grandes, 1 pequeña) se representa debajo:



Pedrito entonces se pregunta: ¿cuántos nodos hay en el árbol asociado con la configuración de la botella (l, s)

Entrada

La entrada comienza con un número T que indica el número de casos de prueba. Cada caso de prueba contiene 2 números l,s que indican la configuración inicial de la botella.

Restricciones/Consideraciones

$1 \leq, T \leq 10000$

$1 \leq, l, s \leq 1000$

Salida

La salida consiste en el número de nodos generados en el árbol. Dado que la respuesta puede ser muy grande debe imprimirlo modulo 9999959999.

NOTA: Para implementar su solución tenga en cuenta que:

$(a + b) \bmod n = (a \bmod n + b \bmod n) \bmod n$

Ejemplo de Entrada	**Ejemplo de Salida**
5 2 1 6 5 100 2 19 78 1000 1000	21 31654 5306431377 1942584859 4124225148