**Proyecto Discretas 2 – Documento Explicativo**

Este programa simula la evolución de clanes mágicos y calcula las particiones enteras del tamaño de cada clan. Permite:

* Agrupar espíritus en clanes mediante una estructura de **conjuntos disjuntos (Union-Find)**, usando operaciones de unión.
* Calcular, para cada clan, cuántas formas distintas existen de distribuir su energía, es decir, contar las **particiones enteras** del tamaño del clan, mediante una función recursiva con memorización.

**Funciones:**

**inicializar(n: int) -> tuple[list[int], list[int]]**

Inicializa las estructuras de datos para los clanes, donde cada espíritu empieza en su propio clan.

**Parámetros:**

* n: Número total de espíritus.

**Return:**

* Una tupla con dos listas:
  + padre: donde padre[i] es el representante del clan del espíritu i.
  + tamano: donde tamano[i] es el tamaño del clan cuyo representante es i.

**buscar(padre: list[int], x: int) -> int**

Encuentra el representante (líder) del clan al que pertenece el espíritu x, utilizando compresión de caminos para optimizar futuras búsquedas.

**Parámetros:**

* padre: Lista con los representantes de cada elemento.
* x: Espíritu a consultar.

**Return:**

* Índice del líder del clan de x.

**unir(padre: list[int], tamano: list[int], x: int, y: int)**

Une los clanes a los que pertenecen los espíritus x e y, si son diferentes, actualizando el representante y el tamaño del clan resultante.

**Parámetros:**

* padre: Lista con los representantes actuales.
* tamano: Lista con los tamaños actuales de cada clan.
* x, y: Espíritus cuyos clanes se quieren unir.

**contar\_particiones(num: int, memo: dict[int, int]) -> int**

Calcula el número de particiones enteras positivas del número num, aplicando la fórmula de Euler para particiones, usando memorización para optimizar el cálculo.

**Parámetros:**

* num: Número entero positivo cuyo número de particiones se desea calcular.
* memo: Diccionario para almacenar resultados parciales y evitar cálculos repetidos.

**¿Cómo funciona?**

* Se utiliza la fórmula generatriz de particiones basada en números pentagonales generalizados.
* Se suman y restan recursivamente las particiones correspondientes, alternando el signo.
* El resultado se devuelve módulo 1,000,000,007 para controlar el tamaño del número.

**Return:**

* Cantidad de particiones enteras de num módulo 1,000,000,007.

**main()**

Función principal que gestiona la entrada y salida del programa.

**¿Cómo funciona?**

* Lee el número de casos de prueba T.
* Por cada caso:
  + Lee el número de espíritus n y la cantidad de operaciones m.
  + Inicializa las estructuras de datos.
  + Procesa las operaciones:
    - union x y: une los clanes de x e y.
    - partitions x: calcula y muestra el número de particiones enteras del tamaño del clan al que pertenece x.

**Ejemplo de Entrada y Salida:**

**Entrada:**

1

5 5

union 1 2

union 2 3

partitions 1

union 4 5

partitions 4

**Salida:**

3

2