**G - Factorial vs Power**

**1. Reducción del Problema  
Descripción en palabras**: Se comparan dos secuencias numéricas, el factorial f(n)=n! y una potencia g(n)=a^n. Se busca el menor entero positivo n tal que f(n)>g(n) para un entero a>1.

**Descripción en lenguaje matemático**: Encontrar el menor n∈Nn \in \mathbb{N} tal que n!>ann! > a^n, con a>1a > 1.

**2. Reducción de la Solución  
Descripción en palabras**: Para cada valor de a, se evalúan sucesivos valores de n hasta encontrar el primero que cumpla n!>a^n.

**Descripción en lenguaje matemático:** Para cada a, hallar el mínimo n tal que ∃n ∈N: n! > a^n.

**3. Código Realizado y Análisis**

* Enlace(s) Código: <https://vjudge.net/solution/61043725/OdvSqqtLyZEInzmA76uj>

**4. Casos de Prueba**

**Caso simple**

Entrada: 2

Salida: 4

**Justificación:** Para a = 2, el n más pequeño donde n! > 2ⁿ es 4 (ya que 4! = 24 > 16 = 2⁴).

**Otro caso un poco más complejo**

Entrada: 3

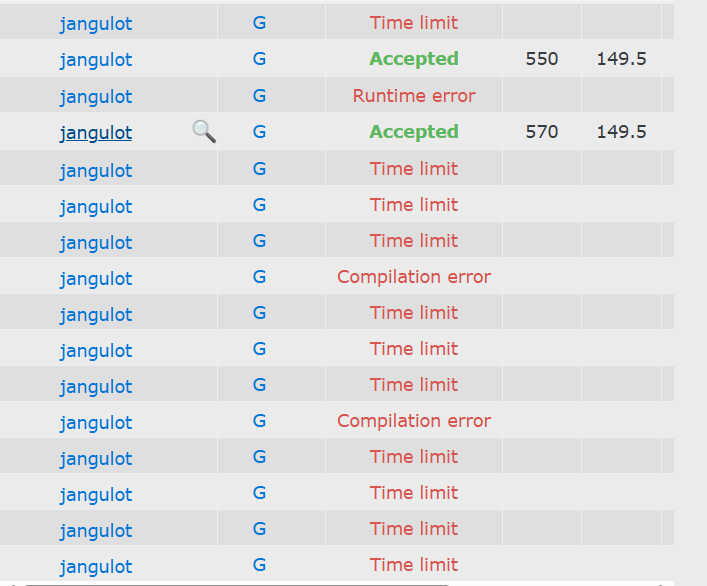
Salida: 7 **Justificación:** Para a = 3, el n más pequeño donde n! > 3ⁿ es 7 (ya que 7! = 5040 > 2187 = 3⁷).

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

**Código incorrecto:** [**https://vjudge.net/solution/61043509/W3qLfP83l7L12YvGaK0T**](https://vjudge.net/solution/61043509/W3qLfP83l7L12YvGaK0T)

* **Descripción del error**: el error principal era cómo hacer el código más eficiente

y creo que hice muchos intentos



* **Proceso de depuración:** Intente muchas cosa pero ninguna funciono, entonces yo entre a esta pagina <https://www.spoj.com/problems/FACVSPOW/en/> donde estaba en mismo problema lei los comentarios como poner n=1000000 y más cosas pero igual no funcionaba entonce fui a los status y mire que muchos lo habían resuelto este ejercicio en java, así que decidí pasar el codigo a java y funcionó.
* **Solución corregida:** <https://vjudge.net/solution/61043725/OdvSqqtLyZEInzmA76uj>

**6. Preguntas de Aprendizaje**

* Temas aplicados:
* crecimiento asintótico de funciones (factorial vs. exponencial).
* Aproximación de Stirling para factoriales.
* Logaritmos para evitar desbordamiento numérico.
* Búsqueda binaria para optimizar la solución.
* Pre Computación de sumas de logaritmos (si se requiere mayor eficiencia).
* Uso de logaritmos naturales (ln) para comparaciones seguras con números grandes.
* Dificultad de la implementación: Alta, especialmente por la parte de eficiencia
* Recursos utilizados: Esta pagina <https://www.spoj.com/problems/FACVSPOW/en/> me ayudo por los comentarios

**7. Feedback LLM**

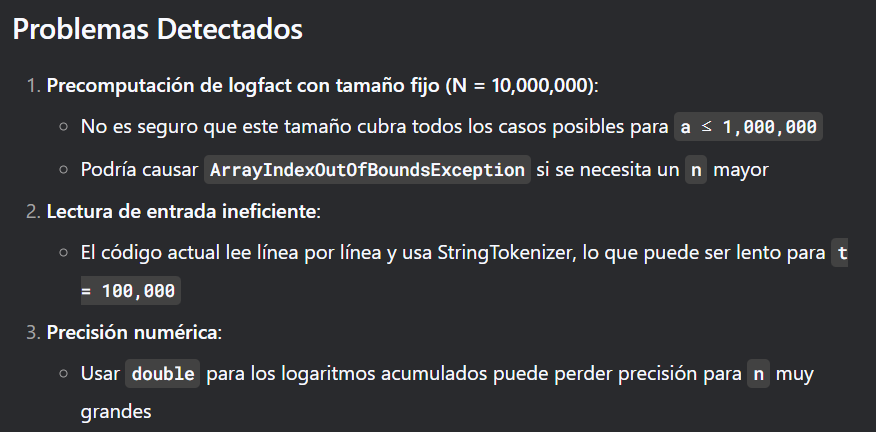
* Envío código LLM: <https://vjudge.net/solution/61083850/ggjIsueMkeMecXRHd4Nz>

**Comparación de su códigos:**

El código del LLM(en este caso Chagpt): es muy simple de hecho se parece a mis primeras version del codigo pero igual sigue teniendo el error del del tiempo excedido, en cambio el mio usa funciones de aproximación de strings para que sea mas eficiente asi que el mio seria mejor porque simplemente superó los test de vjudge

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

me dijo esto



Y la verdad no estoy de acuerdo yo puse N = 10000000 porque en los comentarios mire que ese es el test máximo que hace el juez en línea y para este ejercicio es suficiente