During testing, my bot generally performed well with small numbers and low iterations but consistently failed as calculations became more complex. The primary failure pattern emerged when using n-values above 5 and iterations beyond 6, with the first notable failure occurring at (5,5). Beyond the 4th iteration, the model almost always produced incorrect results, particularly when numbers reached extremely high magnitudes.

The core issue lies in how GPT-4o processes mathematical calculations. Unlike traditional programming languages that compute exact numerical values, GPT does not inherently perform arithmetic but instead predicts answers based on learned patterns from its training data. This means that for sufficiently small calculations, where it has encountered similar values before, it can generate the correct result. However, when numbers grow exponentially, such as in iterative exponentiation, it begins to approximate or guess rather than compute the precise result.

At this point, the model provides an incorrect response, likely due to floating-point estimation errors, token length limitations, or misalignment with training data. The problem worsens at $n \geq 6$, as exponential growth quickly exceeds any dataset the AI has meaningfully encountered.

To achieve better results, you could fine-tune the model or use a mathematics-based approach rather than relying on GPT language-driven pattern recognition. Fine-tuning the model would involve training it on a custom dataset containing correct calculations for a wide range of n-values and iterations, helping it recognize numerical patterns more accurately. However, this approach has limitations since GPT-4o is not inherently a computational model and would still struggle with extreme exponential growth beyond its training data.