Comentarios a *Problems when dealing with big integers*

Comentarios de la subsección *Challange-response verification*

- En el ejemplo que se es muy necesario aquello que se comenta en el siguiente parágrafo que sigue al ejemplo. Recompensar a Alice si Bob ha dado `C` y `B` mal. En ese caso el incentivo será doble, la gente tendrá incentivos para no mentir (ser penalizados) y paralelamente a eso estará incentivados ha trabajar de más por las recompensas.
- Suponiendo un sujeto con voluntad de sabotaje podrida saturar la plataforma si comienza a enviar infinitud de peticiones de corrección. Si las envía todas de forma directa solo se le podrá penalizar un tiempo después, cuando haya ralentizado todo el sistema por el exceso de verificaciones, y quizás después, en el momento de penalizarlo la deuda excede sus bienes y puede saturar el sistema impunemente. Una solución a esto, suponiendo que pueda suceder que tampoco tengo claro si ya tenéis algún mecanismo de defensa contra este tipo de ataques, puede ser retener una parte de su GAS en pretexto de no poder hacer infinitas solicitudes de comprobación, que posteriormente se devolverá si no hay maldad en sus demandas de verificaciones. Fíjate que esto puede también puede constituir un incentivo triple a no *jugarse-la* con aproximaciones numéricas.

Comentarios de la subsección *Modular multiplication & addition*

- Como estamos en $\mathbb{Z}_p$ muchos de los cálculos ejecutados son equivalentes a otros, quizás (no lo sé) puede ser interesante guardar aquellos resultados que se han verificado, así al aumentar el número de operaciones o usuarios en la plataforma o en un *smart contract* concreto pueda tener esto un impacto positivo secundario a la dificultar añadida de aumentar los usuarios o operaciones a realizar. A la larga inclusò el propio *sistema* tienda a optimizarse por el solo. Aunque esta mejora depende fuertemente de que guardar todos esos datos de operaciones verificadas para ser recicladas no sea un problema.
- Esto es solo un pequeño juego matemático que se me ocurrió para números pequeños. Imaginemos que estamos en $\mathbb{Z}_p$, supongamos $\mathbb{Z}_7$ sin perdida de generalidad y queremos calcular $5 \cdot 6$ pero somos incapaces de efectuar el calculo o bien de poder pensar en un número más grande que 7. Tenemos que $5 \cdot 6 \equiv 5 \cdot -1 \equiv -5 \equiv 7$ y no nos hemos pasado de siete.
- Un algoritmo de multiplicación interesante puede ser el Algoritmo de Karatsuba, a diferencia del clasico que tiene complejidad $\mathcal{O}\left(n^2\right)$ este tiene $\mathcal{O}\left(3n^{\log_2(3)}\right)$, donde logaritmo en base dos de 3 es aproximadamente $1.584962500\ldots$ osea que es como $\mathcal{O}\left(3n^{1.585}\right)$. Puedes mirar más en Karatsuba algorithm