Universidad Nacional de Rosario
Instituto Politécnico superior
Departamento de Informática
Taller de programación

1 Modelo de *blockchain* federada

La tecnología *Blockchain* consiste de una cadena de nodos conectados cada uno de los cuales tiene una identificación única. A lo largo del TP asumiremos que cada nodo contiene

- una referencia al bloque anterior,
- un mensaje y
- un número único a nivel sistema que llamaremos id o *hash*.

Una *blockchain* federada es una en la que participan múltiples *blockchain* que coordinan para poder garantizar trazabilidad como un conjunto. Las *blockchain* en estos sistemas se identifican de manera única.

Cada *blockchain* se puede implementar de forma distribuída usando de manera subyacente una red *peer-to-peer* (P2P) en donde cada unidad de cómputo hace de receptor y emisor a la vez. Concretamente, cada unidad de cómputo contiene una copia de un programa *blockchain* y acceso a una parte de la red de nodos.

Cada nuevo nodo que se quiera agregar a la red debe ser consensuado con el resto de los nodos y la garantía de que un nodo es aceptado consiste en la asignación de un número único a nivel sistema. En las implementaciones reales el alta de un nodo puede fallar pero en este trabajo no: otorgaremos un identificador único de manera centralizada.

Es común escuchar que los sistemas *blockchain* son seguros porque requieren del 'visto bueno' de diferentes componentes del sistema para garantizar que las modificaciones sean íntegras. Esto se debe a que cada vez que se modifica un nodo en una *blockchain*, todos los nodos a partir de ese deben ser reverificados contra toda la red. Dicho de otra forma, cada vez que se modifica el nodo iésimo en una blockchain todos los nodos a partir del 'i' deben ser revalidados. La estructura de datos que se utiliza para esta validación se suele llamar *Merkle Tree* (árbol de Merkle) ó *Hash tree* (árbol de hashes).

Árbol de validación La pieza que se encarga de garantizar la integridad de todo el sistema, el árbol de validación, depende de propiedades matemáticas que no se van a abordar en el trabajo aunque se presenta una simplificación de su modelo. El árbol de validación contiene información que verifica qué nodos hasta el momento fueron aprobados dentro de la red y se construye a partir de sus hojas.

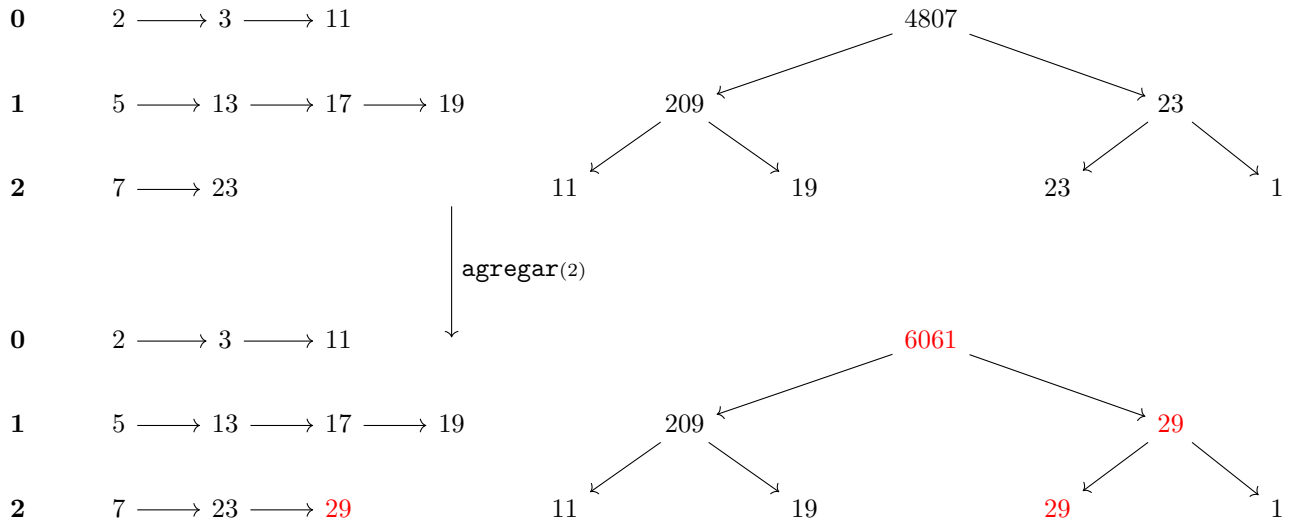
Cada hoja referencia el nodo final de cada *blockchain*, en concreto, tiene de dato el identificador único del último nodo agregado en una *blockchain*. Cada nodo interno del árbol contiene de dato la multiplicación de los datos de sus hijos. Si la cantidad de hojas (cantidad de *blockchains*) no es múltiplo de 2 entonces se completará su estructura hasta alcanzar un árbol completo con 1.

Identificadores de nodos Para otorgar una identificación a un nodo utilizaremos números primos. Cada nuevo nodo a agregar o a revalidar tendrá un número primo distinto y mayor a todos los anteriores. Para acceder a distintos números primos se debe usar el módulo provisto de números primos.

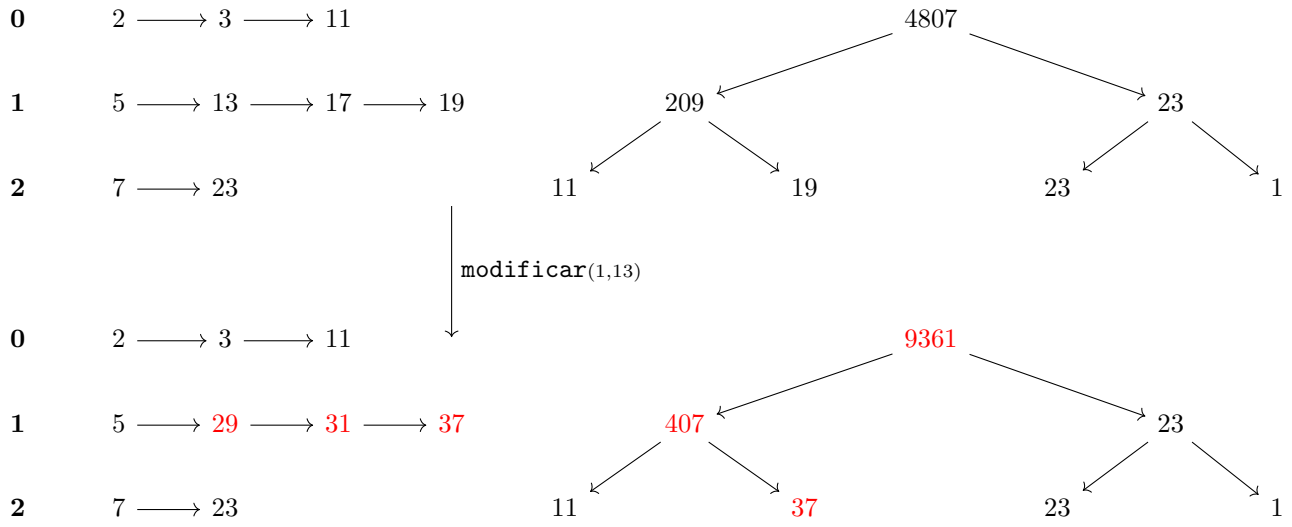
Teniendo en cuenta que el árbol de validación contiene en sus hojas números primos distintos y que los nodos internos se construyen a partir de la multiplicación de sus hijos entonces la raíz del árbol será la multiplicación de todas las hojas y por el teorema fundamental de la aritmética éste será un número único y por lo tanto servirá de validación de todos los datos de la *blockchain* federada.

2 Operaciones sobre una blockchain federada

La operación de **alta** toma la *blockchain* a la que se quiere agregar un nodo y un mensaje y agrega un nuevo nodo con dichos datos a la estructura de *blockchains* y actualiza el árbol de validación.



La operación de **actualización** toma la *blockchain* en la que quiere modificar el nodo, el identificador único del nodo, el nuevo mensaje y actualiza su identificador y todos los subsecuentes identificadores. Además, refleja los cambios en el árbol de validación.



La operación de **validación** verifica que los nodos agregados a la red *blockchain* federada se correspondan con el árbol de validación actual. Esto implica verificar que

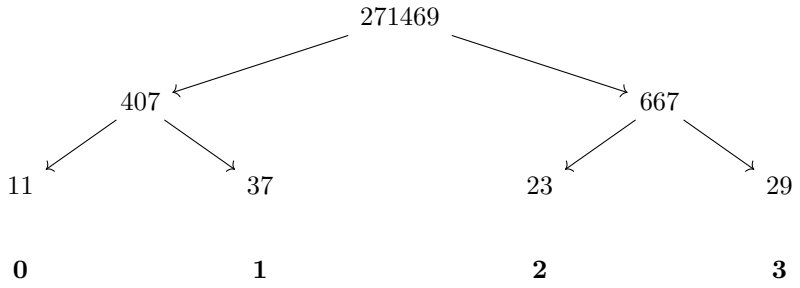
- cada identificador dentro de una *blockchain* sea mayor al anterior y
- la raíz del árbol de validación coincide con la multiplicación de los identificadores de los nodos finales de cada *blockchain*.

Nota: asumimos para la validación que todos los identificadores de nodos son números primos.

Supongamos ahora que el dominio de diferentes sectores de la *blockchain* federada depende de diferentes naciones entre las cuales se encuentran Argentina y Suecia. Supongamos que a Suecia se le corrompe su copia del árbol de validación por lo que nos pide ayuda. Suecia nos debería dar la raíz de su árbol de validación junto con los índices que representan sus *blockchains* en la *blockchain* federada. Con esta información podríamos validar si la raíz que creen valida sus *blockchains* es correcta.

La operación **validar_subconjunto** toma un número que representa la validación esperada de un conjunto de *blockchains* y un par de números que representan mínimo y máximo identificador de un conjunto de *blockchains* y devuelve 1 en caso de que la validación del conjunto coincida con el valor esperado pasado como primer argumento.

Por ejemplo, si las *blockchains* de Suecia son {0,1} entonces el número de validación debe ser 407 mientras que si son {0,1,2}, el número de validación debe ser 9361.



3 Ejercitación

- Proponga una o varias estructuras de datos para representar una red federada de *blockchain*. Tenga en cuenta que su propuesta debe incluir la representación concreta de las *blockchain* además del árbol de validación.
- Implemente las operaciones de alta, actualización, validación y validación de conjunto.
- Incluya un `main.c` en el que se muestre un caso de uso para la *blockchain*.
- Elabore un informe de máximo 1 página **manuscrito** en donde se expliquen las decisiones más importantes tomadas a lo largo del trabajo. Este informe se entregará el último día de clases antes de la entrega del trabajo.

4 Evaluación

Serán parte del criterio de evaluación:

- la elección y correcta implementación de las estructuras de datos,
- la correcta implementación de las operaciones,
- la claridad y relevancia del ejemplo de uso propuesto,
- la calidad de código de todo el proyecto y su correcta estructuración, modularización y diseño,
- la claridad del informe presentado y
- el desempeño individual en la defensa del trabajo.

Toda línea de código entregada se considera de su autoría y por lo tanto debe poder ser defendida en una jornada destinada a tal fin.