3INDF BlocNote activité 1 : enchaînement de blocs (blockchain)

But : comprendre comment enchaîner des blocs dans une chaîne de blocs.

1.1 Théorie

Une **chaîne de blocs** (blockchain) est un registre de données public, inaltérable et (presque) inattaquable. Les **blocs** pérennisent des informations, comme par exemple des transactions financières. Les deux exemples les plus connus de chaînes de blocs sont Bitcoin et Ethereum.

Dans une chaîne de blocs, chaque bloc a un numéro d'identification unique. Le premier bloc s'appelle **bloc genèse**, et le prochain bloc est nécessairement enchaîné au bloc-genèse, puis les blocs suivants peuvent être attachés à n'importe quel bloc de la chaîne, dont le bloc-genèse. Le bloc auquel un nouveau bloc est enchaîné s'appelle **bloc-parent**.

Les **mineurs** sont les entités (ordinateur, serveur, super-ordinateur, élève) qui tentent d'insérer un nouveau bloc dans la chaîne de blocs en trouvant la solution d'un puzzle mathématique, appelé **cible**. Ce travail s'appelle **minage,** et le résultat est la **preuve de travail (PoW)**. La cible du puzzle dépend des caractéristiques du bloc-parent. Le mineur qui réussit à insérer un bloc reçoit une **récompense** sous forme de **cryptomonnaie**. La récompense est le seul mécanisme par lequel de la cryptomonnaie est créée dans une chaîne de blocs.

Il est possible que des mineurs attachent plusieurs blocs à un même bloc-parent, et que d'autres blocs s'attachent à ces blocs, créant plusieurs fourches dans la chaîne de blocs. Pour éviter que la cryptomonnaie ne soit dépensée dans plusieurs fourches (problème de **double-dépense**, « double-spending »), il faut que tous les participants s'entendent pour déterminer quelle fourche est la bonne. Le **consensus** consiste à reconnaître la fourche de blocs la plus longue comme étant la chaîne de blocs officielle.

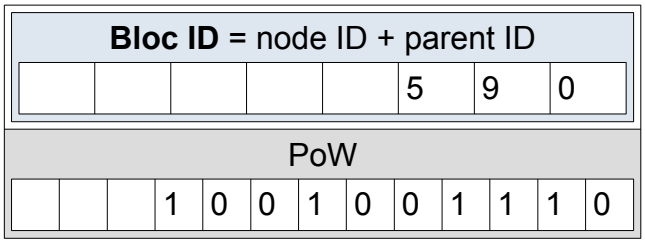
Par conséquent, étant donné que seule la fourche la plus longue compte, les blocs sur les autres fourches ne rapportent rien.

1.2 BlocNote

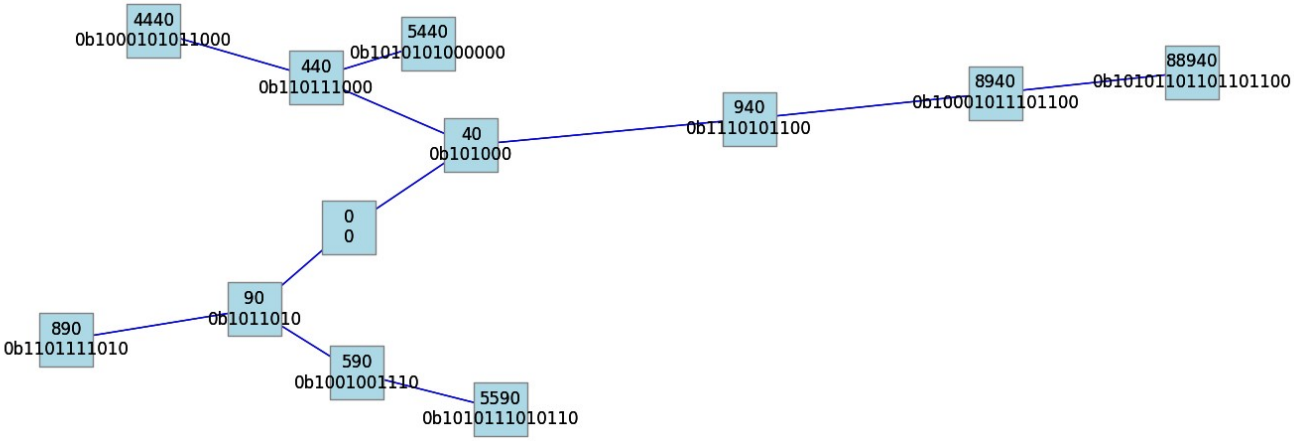
BlocNote est un modèle de chaîne de blocs développé pour les élèves de la maturité gymnasiale qui reprend les principes de Bitcoin et Ethereum.

Dans BlocNote, chaque bloc possède un numéro d'identification unique, le **blocID**, composé du numéro de mineur (nodeID) suivi du blocID du bloc-parent. Pour insérer un bloc dans la chaîne, le mineur choisit d'abord un bloc-parent, puis calcule la PoW, qui consiste à déterminer la notation binaire du blocID.

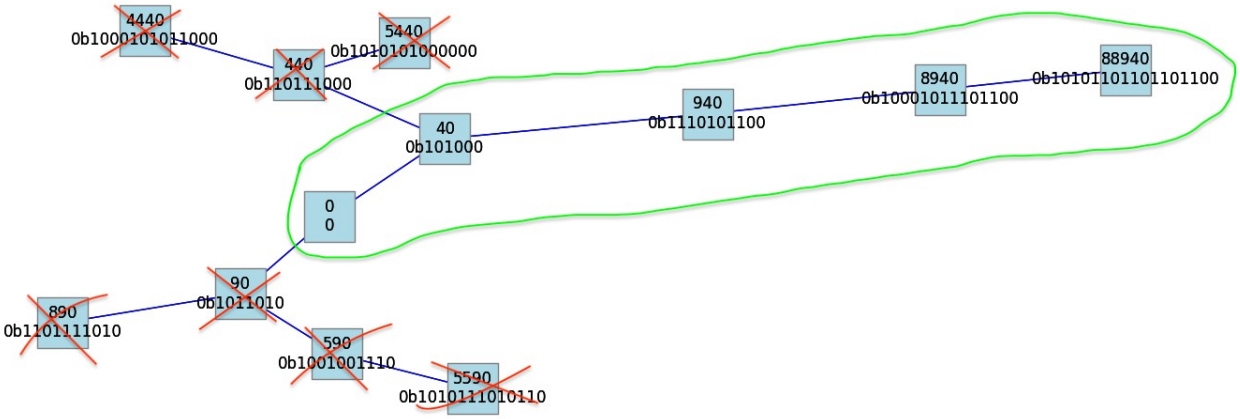
Exemple : le nœud 5 enchaîne un bloc au bloc 90, le blocID est donc 590 et la PoW est 1001001110 (car 590 = 512 + 64 + 8 + 4 + 2).



Le graphique ci-dessous montre un exemple de chaîne de blocs, où les blocs contiennent leur numéro d'identification unique et la PoW. Etant donné que les mineurs peuvent enchaîner les nouveaux blocs à n'importe quel bloc-parent, plusieurs fourches se développent.



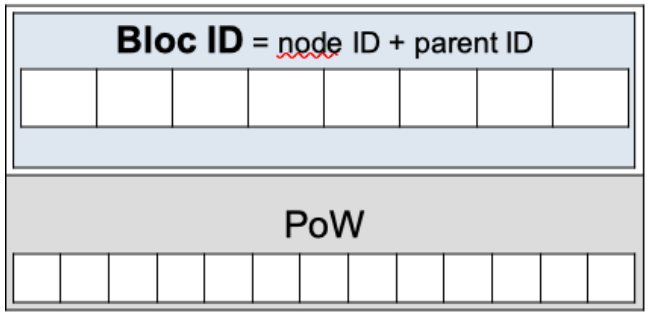
A la fin de l'activité, les mineurs reçoivent une récompense de 10 R$ par bloc inséré dans la fourche la plus longue, tous les autres blocs ne rapportent rien. Dans l'exemple ci-dessous, les noeud 4 et 9 reçoivent 10 R$ chacun, le noeud 8 reçoit 20 R$, mais le noeud 9 ne reçoit rien.



1.3 Activité

Matériel :

- panneau d'affichage (tableau blanc et aimants, ou panneau en liège et punaises, ou poster et scotch) pour la chaîne de blocs ;
- ordinateur pour les programmes « rousseau_interac.py » pour vérifier les nouveaux blocs, et « rousseau_visualization.py » pour visualiser la chaîne ;
- feuillets :



- papier brouillon (pour les calculs), crayon, calculatrice.

Déroulement :

1. les élèves travaillent par groupes de 3 ou 4, chaque groupe représente un mineur
2. chaque mineur reçoit un numéro de mineur (nodeID) entre 1 et 9, et le garde secret.
3. la chaîne de blocs commence par le « bloc genèse », dont le « bloc ID » est 0.
4. Pour insérer un bloc, le mineur effectue les opérations suivantes :
   a) choisir un bloc-parent sur lequel enchaîner le nouveau bloc,

      Attention : n'importe quel mineur peut attacher un bloc à n'importe quel bloc-parent, mais un mineur donné ne peut pas attacher plusieurs blocs au même bloc-parent, car le blocID doit rester unique.

   b) déterminer le blocID du nouveau bloc, composé du nodeID du minieur et du blocID du bloc-parent,
   c) prouver qu'il a effectué un travail (PoW) en trouvant la solution du puzzle, qui consiste à calculer la représentation binaire du blocID du bloc-parent,
   d) Le mineur écrit le blocID du nouveau bloc et sa PoW sur le feuillet qui représente un bloc.
5. Les mineurs collent leurs feuillet sur le panneau d'affichage, et dessinent un trait pour montrer à quel bloc-parent chaque bloc est enchaîné.
6. Au fur et à mesure que des blocs sont enchaînés à la chaîne, les mineurs vérifient les blocs minés par les autres nœuds afin de s'assurer de leur validité.

   Attention : tous les blocs enchaînés à un bloc non-valide sont eux aussi non-valides, et ne compteront pas à la fin de l'activité.
7. À la fin de l'activité, les mineurs reçoivent 10 CR pour chaque bloc inséré dans la fourche la plus longue.

1.4 A retenir

Le blocID est composé du nodeID et du blocID du bloc-parent, et la PoW consiste à trouver la notation binaire du blocID : chaque bloc contient donc tout l'historique des blocs précédents de la chaîne. C'est ce qui rend la chaîne de blocs **inaltérable** : pour altérer un bloc déjà inséré dans la chaîne, le mineur malveillant doit modifier son blocID et sa PoW ; or les blocs subséquents comportent le blocID avant qu'il ne soit altéré, le mineur malveillant doit donc aussi modifier les blocIDs et les PoW de tous les blocs subséquents au bloc altéré.

Pendant que le mineur malveillant modifie les blocs, les autres mineurs continuent d'ajouter des blocs, rendant le travail du mineur malveillant encore plus difficile, et contribuent à une fourche probablement plus longue que celle que le mineur est en train d'altérer.

Pour autant que le mineur malveillant ne représente pas plus que 50 % de la puissance de calcul de la chaîne, les chaînes minées par les autres mineurs vont grandir plus rapidement que la chaîne que le mineur malveillant est en train de modifier. Son travail est donc inutile, car sa chaîne ne sera jamais la plus longue.

Les mineurs ont donc plus avantage à utiliser leur puissance de calcul pour insérer des blocs sur la fourche la plus longue, plutôt que de modifier des blocs dans une fourche qui sera très probablement abandonnée. C'est ce qui rend la chaîne de blocs « presque » **inattaquable** : il est plus avantageux pour un nœud de jouer selon les règles que de tenter de contrôler la chaîne.

L'attaque à 51 % (**51 % attack**) menace toutes les chaînes de blocs. Si une collusion de mineurs réussit à réunir 51 % de la puissance de calcul, ils peuvent contrôler la chaîne et s'attribuer toute la cryptomonnaie.

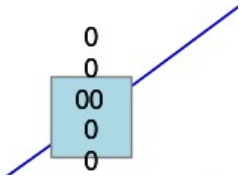
Dans la vraie vie...

Dans une chaîne de blocs basée sur la PoW, comme Bitcoin, le **niveau** de difficulté du puzzle varie afin de ralentir ou d'augmenter la croissance de la chaîne de blocs. Dans Bitcoin, le puzzle consiste à trouver la pré-image d'une fonction de hachage, une opération très complexe qui demande une immense puissance de calcul.

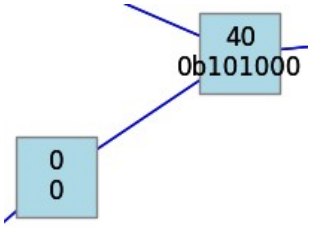
1.5 Exemple complet

L'exemple ci-dessous montre le développement d'une chaîne de blocs BlocNote étape par étape.

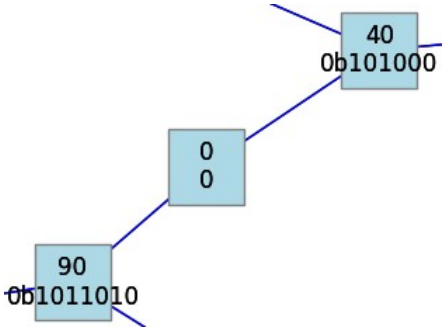
0) Toutes les chaînes commencent par le bloc-genèse:



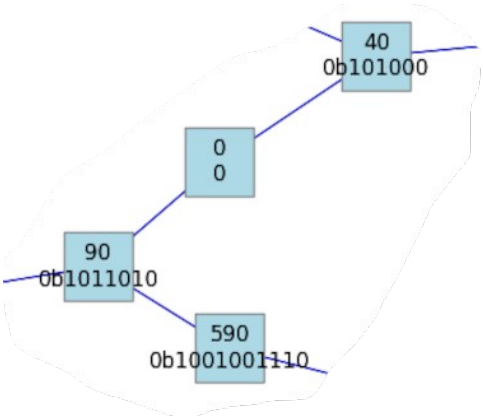
1) Le noeud 4 enchaîne un bloc au bloc-genèse, le blocID est donc 40.



2) Le noeud 9 enchaîne aussi un bloc au bloc-genèse, le blocID est donc 90.

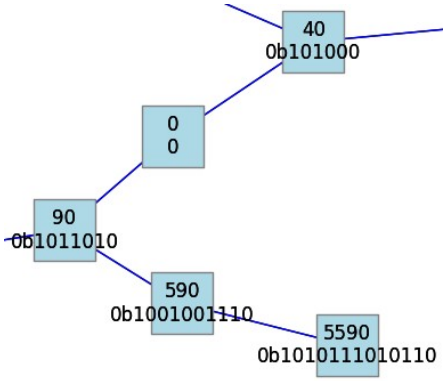


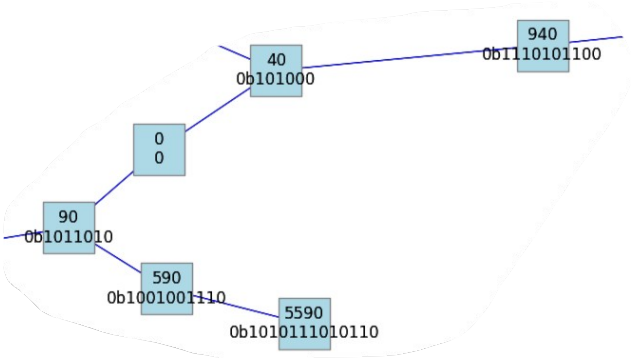
3) Le noeud 5 enchaîne un bloc au bloc 90, le blocID est donc 590.



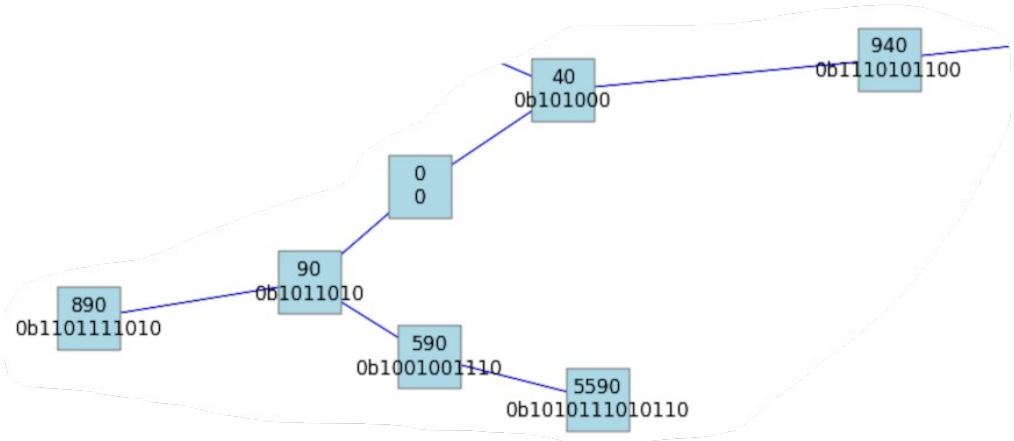
4) Le noeud 5 enchaîne un bloc au bloc 590, le blocID est donc 5590.

5) Le noeud 9 enchaîne un bloc au bloc 40, le blocID est donc 940.

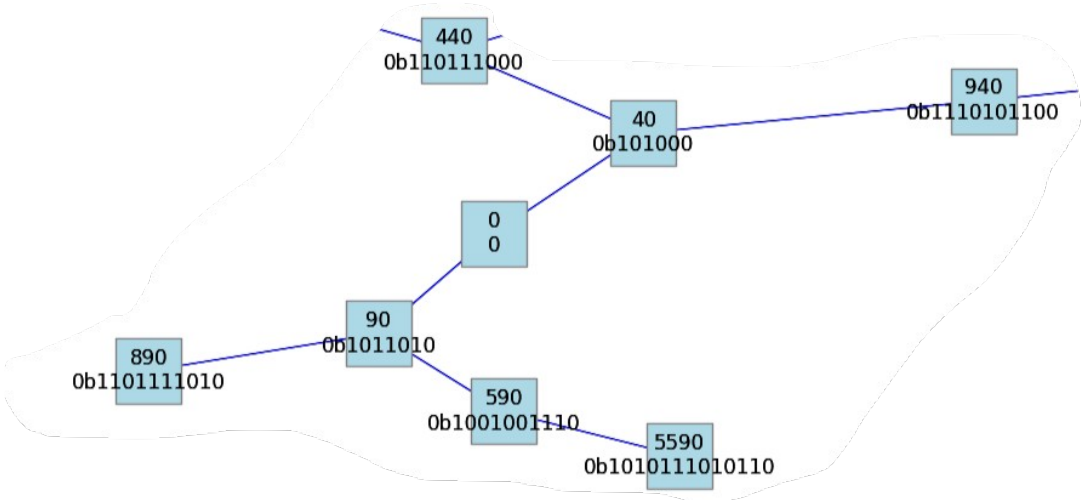




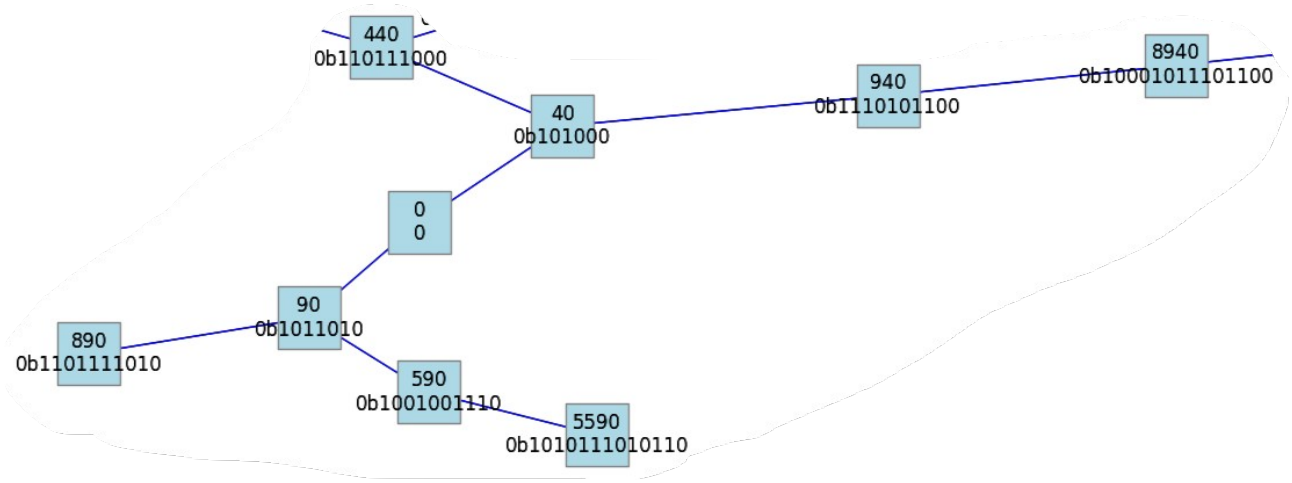
6) Le noeud 8 enchaîne un bloc au bloc 90, le blocID est donc 890.



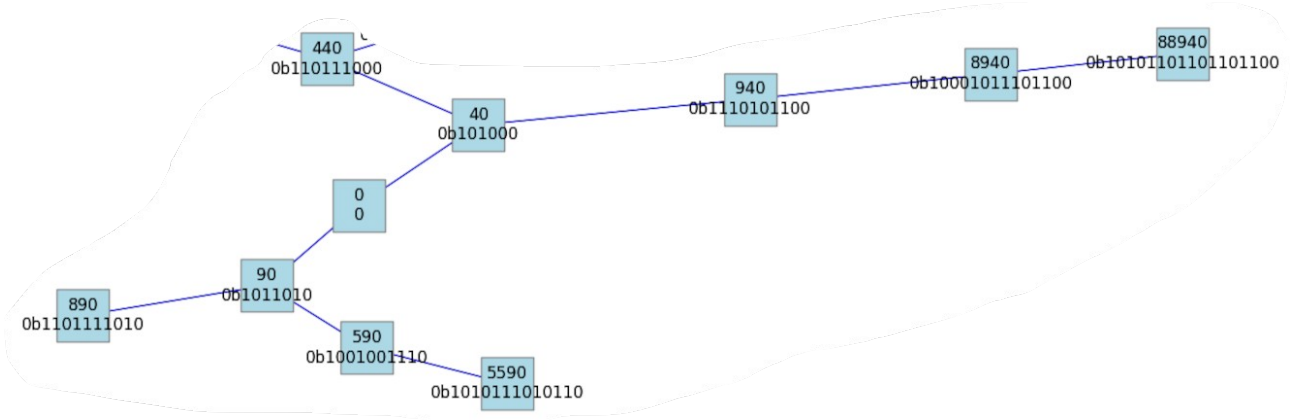
7) Le noeud 4 enchaîne un bloc au bloc 40, le blocID est donc 440.



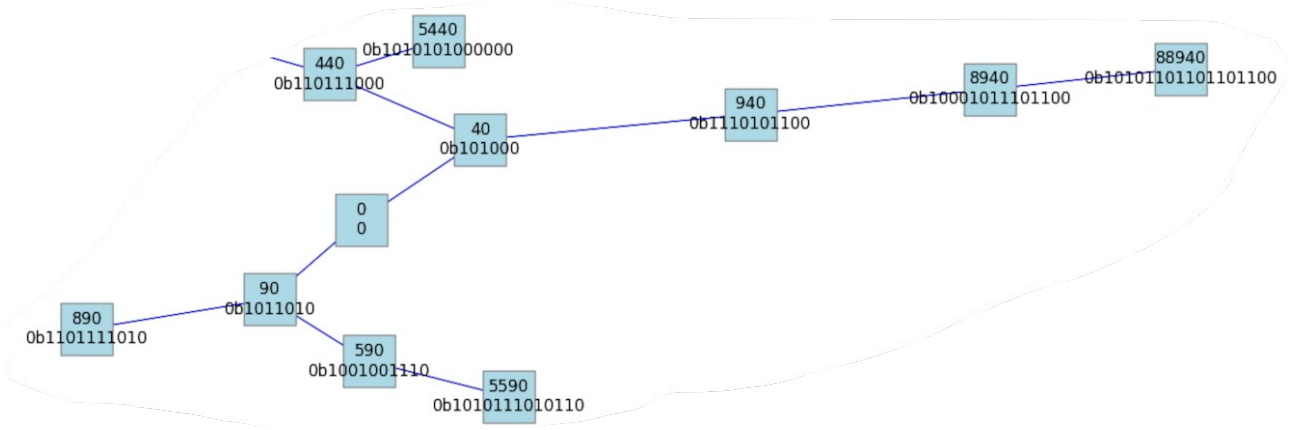
8) Le noeud 8 enchaîne un bloc au bloc 940, le blocID est donc 8940.



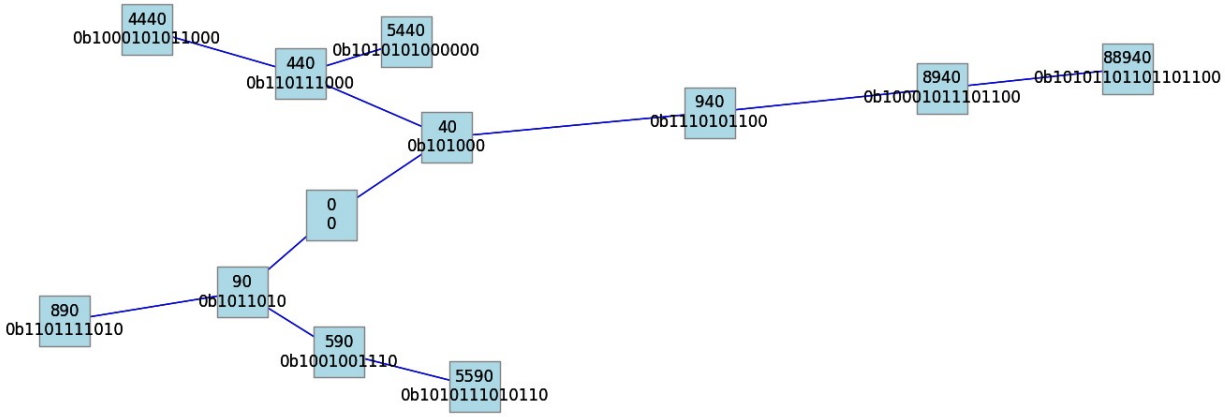
9) Le noeud 8 enchaîne un bloc au bloc 8940, le blocID est donc 88940.



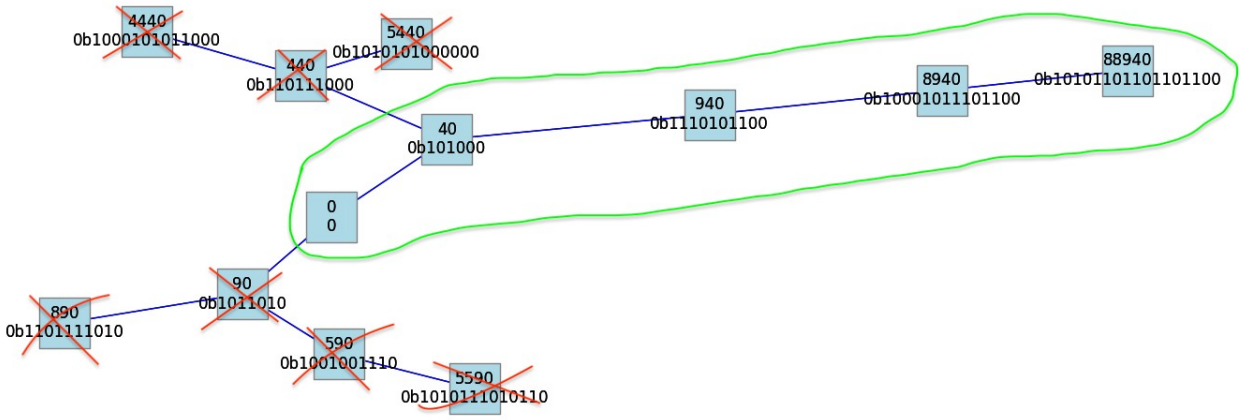
10) Le noeud 5 enchaîne un bloc au bloc 440, le blocID est donc 5440.



11) Le noeud 4 enchaîne un bloc au bloc 440, le blocID est donc 4440.



A la fin de l'activité, seule la chaîne la plus longue compte: les noeud 4 et 9 reçoivent 10 R$ chacun, le noeud 8 reçoit 20 R$, mais le noeud 9 ne reçoit rien.



Dans la réalité, cette représentation graphique n'est pas fournie par la chaîne de blocs, toutes les informations se retrouvent dans un **registre** dans lequel les blocs et les informations qu'ils contiennent apparaissent en ordre chronologique d'enchaînement, comme montré dans le tableau ci-dessous.

Ordre	BlocID	PoW
	0	0
1	40	0b101000
2	90	0b1011010
3	590	0b1001001110
4	5590	0b1010111010110
5	940	0b1110101100
6	890	0b1101111010
7	440	0b110111000
8	8940	0b10001011101100
9	88940	0b101011011101101100
10	5440	0b1010101000000
11	4440	0b1000101011000

Ce sont des services extérieurs (les élèves, dans le cas de BlocNote) qui analysent le registre pour en déduire les informations utiles pour les utilisateurs, comme le solde de cryptomonnaie associée à une adresse donnée, par exemple.

Exemple d'attaque

Par exemple, le mineur 5 souhaite changer le numéro du mineur du bloc 490 pour s'attribuer la récompense. Il modifie donc le bloc 940 en changeant le blocID à 540, pour faire semblant que c'est lui qui l'a miné, et recalcule la PoW pour qu'elle corresponde à 540. Mais il devra aussi modifier le blocID des blocs 8940 et 88940 en 5840 et 55840, et recalculer leurs PoWs.