ESISAR

CS353 - Algorithmique

TP numéro 5 et 6
RAINBOW TABLE

Table des matières

1 Objectifs du TP numéro 5 et 6

Ce TP va permettre l'implémentation d'une RainbowTable (vu en TD).

Pour faire ces 2 TP, vous avez besoin de :

- une séance de 1h30 avec l'enseignant en salle de TP pour commencer
- 4H de préparation entre les 2 TP
- une autre séance de 1h30 avec l'enseignant en salle de TP pour terminer
- une poignée de courage et un zeste de persévérance !

Tous les programmes réalisés seront obligatoirement en Java. Le rendu se fera à l'aide de la grille de réponse disponible sur Chamilo.

2 Les documents du TD

Relisez les documents vus en TD :

https://en.wikipedia.org/wiki/Rainbow_table
http://kestas.kuliukas.com/RainbowTables/
https://en.wikipedia.org/wiki/MD5

3 Exo 1 - La force brute ...

Votre ami Jack Le Hacker a réussi à s'introduire sur un site WEB de e-commerce et à voler la base de données du site.

A partir de cette base de donnée, votre ami vous a fourni une liste de login / mot de passe , de la forme suivante :

Alice	8FC92036B2963C604DC38B2DDB305148
Bob	367F3AC1129CC92DCBB8C4B9EA4EE55A
Clara	38251B4C8C210841C60CDE0B7E4C7A87

La colonne 1 contient le login de l'utilisateur, la colonne 2 contient le hash MD5 du mot de passe de l'utilisateur

Sur ce site WEB, tous les mots de passe sont composés uniquement de chiffres, et ont toujours pour longueur 8. Exemples de mot de passe : 12500123 , 42456123 , … Attention, les mots de passe sont bien des chaînes de caractères, mais composés uniquement de chiffres.

Par une attaque de type brute force, déterminer le mot de passe de tous les utilisateurs.
Indiquez le temps de calcul de tous ces mots de passe par brute force.

Pour vous aider à démarrer, voici ci dessous un exemple de code montrant comment calculer un MD5 en Java et l'afficher dans la console :

```
	public static void main(String[] args) throws NoSuchAlgorithmException
	{
		MessageDigest digest = MessageDigest.getInstance("MD5");
		String str ="12345678";
		byte[] hash = digest.digest(str.getBytes());

		// Affichage
		StringBuilder sb = new StringBuilder();
		for (byte b : hash)
		{
			sb.append(String.format("%02X", b));
		}
		System.out.println(sb.toString());

	}
```

Voici maintenant un exemple montrant comment convertir un nombre en une chaîne de caractères avec un nombre fixe de caractères (ici 8) :

```
            int a=123;
            String str = String.format("%08d", a);
            System.out.println("str="+str);    // Affiche str=00000123
```

Enfin, voici un exemple montrant comment convertir une chaîne de caractères contenant une représentation hexadécimale en un tableau de byte

```
String s = "404142434445464748494A4B4C4D4E4F";

// Conversion de s en un tableau de byte, le premier byte est 0x40 , le deuxième 0x41 , et ainsi de
suite
int len = s.length();
byte[] data = new byte[len / 2];
for (int i = 0; i < len; i += 2)
{
   data[i/2] = (byte) ((Character.digit(s.charAt(i), 16) << 4) + Character.digit(s.charAt(i+1), 16));
}

// Affichage de data
for (int i = 0; i < data.length; i++)
{
       System.out.println("data["+i+"]="+data[i]);
}
```

4 Exo 2 - Calcul d'une chaîne

Faites une fonction

```
public int calculChaine(int px)
```

qui

- prend en entrée un entier quelconque entre 1 et 99_999_999
- convertit ce nombre en une chaîne de 8 caractères (mot de passe PX, exemple « 12012012 »)
- calcule le hash MD5 de la chaîne de caractère
- le hash MD5 est ensuite réduit par la fonction de réduction R0, on obtient le mot de passe P0
- on répète ensuite l'opération : calcul du hash MD5 de P0, puis réduction par R1, on obtient P1
- …
- calcul du hash MD5 de P998, puis réduction par R999, on obtient P999
- la fonctionne retourne ensuite P999

```
   Hash      R0      Hash   R1                    Hash      R999
PX   =>  H0  => P0   =>  H1  => P1 => ...   P998  =>  H998  => P999
```

Vous utiliserez la fonction de réduction suivante. num correspond au numéro de la fonction de réduction (num varie de 0 à 999).

```
private int reduction(byte[] bs,int num)
{
        int res = num;

         int mult = 1;

        for (int i = 0; i <8 ; i++)
        {
              res = res+mult*( (bs[i]+256) % 10);
              mult = mult*10;
        }
        return res % 100_000_000;
}
```

Par exemple si l'entrée de la fonction **`CalculChaine`** est 1, la sortie devra être `43183201.`

5 Exo 3 - Calcul de la table complète et sauvegarde dans un fichier

Faites maintenant un programme qui calcule 1_000_000 chaînes et qui stocke pour chaque chaîne les valeurs (PX et P999) dans une table de hachage à adressage ouvert de taille 1_000_003.

L'algorithme sera le suivant

- faites une boucle de 0 à 999_999
- pour chaque pas de la boucle
  - calculer un nombre aléatoire entre 0 et 99_999_999 (voir le code fourni ci dessous)
  - en déduire PX et P999
  - insérer le couple (PX,P999) dans la table de hachage, **P999 est la clé** et PX est une valeur satellite
  - si la clé est déjà existante dans la table de hachage (il y a déjà dans la table une chaîne se terminant par P999), alors le couple (PX,P999) n'est pas inséré (la chaîne que l'on vient de calculer est oubliée)

Détail sur le calcul des nombres aléatoires

Vous allez utiliser le code suivant pour générer les nombres aléatoires :

```
	Random r = new Random(0);
	int a0 = r.nextInt(100_000_000); // Genere un premier nombre entre 0 et 99_999_999
	System.out.println("A0="+a0);
	int a1 = r.nextInt(100_000_000); // Genere un deuxième nombre entre 0 et 99_999_999
	System.out.println("A1="+a1);
```

Le premier nombre généré est toujours `69741360`, le deuxième `85505948`, et ainsi de suite.
Ceci permet à votre programme d'avoir toujours le même comportement.

Détail sur la table de hachage

Il est ici possible de faire quelque chose de très simple, car il n'est pas nécessaire de gérer les éléments supprimés.

Vous pouvez utiliser la classe suivante

```
public class Noeud
{
	public int px;
	public int p999;
}
```

et la table sera un tableau de Noeud

```
Noeud[] table = new Noeud[10061];
```

Si l'élément i est vide, alors table[i] est égal à null.

Questions :
1/ Combien de couple (px,p999) sont stockés dans votre table de hachage ?
2/Quel est le temps de calcul de votre table ?
3/ Pourquoi la table de hachage a une taille de 1_000_003 ?

Une fois votre table de hachage calculée, sauvegardez la dans un fichier (il suffit d'écrire tout le tableau dans un fichier).

6 Exo 4 - Utilisation de la table pour « casser » un mot de passe

Faites maintenant un programme :

- charge votre table de hachage en mémoire à partir du fichier
- demande en entrée un hash MD5 de mot de passe (le mot de passe est toujours 8 caractères numériques)
- retourne le mot de passe en clair

Indiquez le temps de calcul d'un mot de passe.
Faites une démonstration à l'enseignant.