W12 手寫功課

劉至軒

June 8, 2019

Problem 1.

1. 令四個花色（和 Joker）分別為 1，10，100 10000。則對於一張卡 c，我們定義 $H(c)$ = 花色對應的值加上其數字（對於 Joker，第一個是 1，第二個是 2）。

2. 還是用程式比較清楚... `n`為目前的根節點，而`n->l`、`n->r`為指向其左右子樹的指標。

```
1  string getHash(Node n){
2    if(!n) return "0";
3    if(n->l && n->r){
4      return "4" + getHash(n->l) + getHash(n->r);
5    } else if(n->l){
6      return "3" + getHash(n->l);
7    else if(n->r){
8      return "2" + getHash(n->r);
9    } else {
10     return "1";
11   }
12 }
```

則題目中的二元樹的雜湊值分別為：$0, 1, 31, 21, 411$。

Problem 2.

1. 對於每一個字串，都必須要有不同的雜湊值，也就是需要數有幾個相異字串。答案：

$$\sum_{i=1}^{6} 26^i$$

個不同的值。

2. 沒辦法，因為可能還需要不知道的資訊，譬如一個 Key 等，例子：Viginere Cipher 中的 Key 是解碼的必要條件，如果沒有的話只能慢慢猜，非常難回推其他的密文。

3. 攻擊者可以透過類似 Tampermonkey 等具有類似功能的工具 Intercept 到被攻擊者的電腦傳輸到伺服器的雜湊過後的函數，而因為此雜湊不具有 $One - wayness$，Intercept 之後就可以容易復原原本的密碼／認證機構，而獲得權限。

4. 令第 i 個輸入為 $a_i = i \cdot 1000000007$，則全部都會跑進去餘數為零的 Bucket 裡面，到最後查詢會變成 $O(N)$。

Problem 3.

1. 程式如下：

```
int M, C; //given values
int hash(string s){
  int currentPow = 1, res = 0;
  reverse(s.length(), s.end());
  for(char c : s){
    res = (res + (c - 'a') * currentPow) % M;
    currentPow = C * currentPow % M;
  }
  return res;
}
```

   此程式對於每一個字元都跑 $O(1)$ 的運算，所以時間 $O(1) \times n = O(n)$。

2. $s_1 = a$、$s_2 = aa$、$s_3 = aaa$，此處 $H(s_1) = H(s_2) = H(s_3) = 0$。

3. 答案：

$$y = C \times (x - s_l \cdot C^{k-1}) + s_{r+1} \pmod{M}$$

4. 先計算 s 的雜湊值 $H(s)$，花費時間 $O(n)$，然後再對於每一個長度為 n 的 t 的子字串，都可以轉移：先 $O(n)$ 計算 $H(t[1,n])$ 的值，然後轉移到下一個狀態（$H[2,n+1]$），到 $H(t[m-n+1,m])$ 為 $O(1)$，然後在這個步驟總共花了 $O(m-n)$。總複雜度 $O(n+(n+m-n)) = O(n+m)$。