

# 中央大學 112 學年度計算機概論 II 作業 03

## Problem 03:駭客入侵

### 問題描述

XXX 是一個國際型駭客組織，他們即將發起一波駭客行動。作為駭客組織的一員，你被要求主導這個行動，請編寫一個入侵系統來模擬駭客行為，已知電腦資料的儲存與病毒入侵的規則如下：

#### 1. 電腦資料儲存

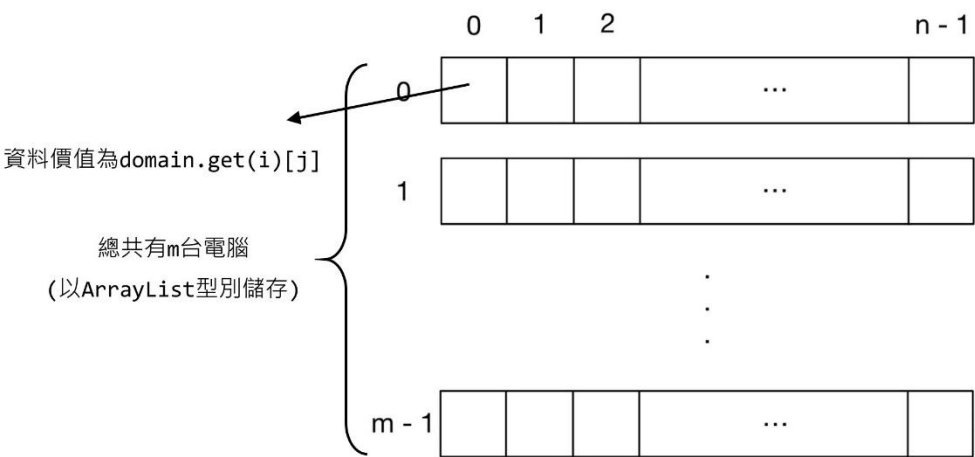
在相同網域中，每台電腦具有相同數量的資料，以陣列 `data[]` 的方式儲存，其中 `data[i]` 可為整數或 0，分別代表不同意義：

- `data[i]` 為整數：該電腦中第 `i` 筆資料價值為 `data[i]`
- `data[i] = 0`：若 `data` 為一般用戶電腦(被入侵者)，則代表第 `i` 筆資料為加密資料，駭客病毒無法入侵；若 `data` 為駭客電腦(入侵者)，則代表第 `i` 筆資料為可能的病毒，具體駭入方式請參照第三點駭客入侵。

#### 2. 相同網域電腦資料

當多台電腦以 `ArrayList` 形式儲存於相同網域時，每台電腦的資料數量會相同(即 `data.length` 相同)，而這些電腦組成的網域即為一個型別為 `ArrayList<int[]>` 的串列 `domain`，其中 `domain.get(i)[j]` 代表

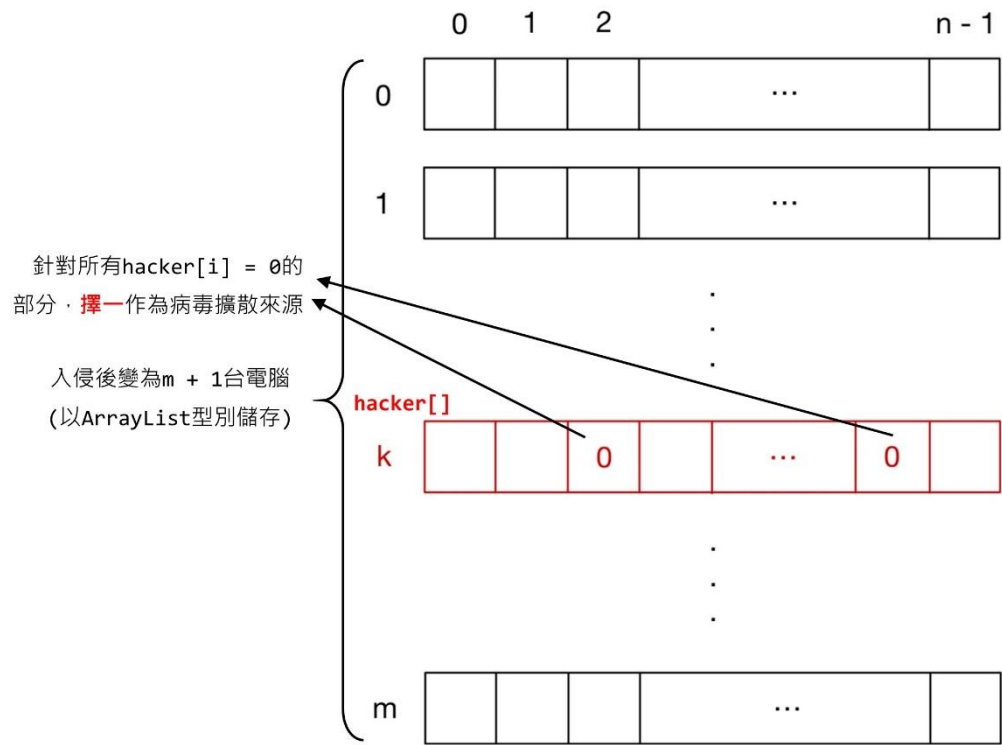
第  $i$  台電腦的第  $j$  筆資料價值，如下圖一所示。



圖一

### 3. 駭客入侵

當駭客以電腦 `hacker[]` 入侵網域 `domain` 時 (`hacker.length` 必等於  $n$ )，系統會將 `hacker[]` 陣列插入 `domain` 中，如下圖二所示。



圖二

當駭客的電腦 `hacker[]` 進入網域 `domain` 後，將選取 `hacker[]` 陣列中

的其中一個 `hacker[] = 0` 做為病毒的起始點(其他的 `hacker[] = 0` 定義仍代表加密資料，不可侵入)，此時病毒將從起始點開始入侵網域 `domain`，獲取 `domain.get(i)[j]` 的資料，其中 `domain.get(i)[j]` 可包含 `hacker[]` 陣列，且最終病毒行進路線不可有交集。

請注意，病毒的入侵可隨時停止，不一定要走到最底部。

## 實作限制

請實作以下 3 個 function，功能說明如下：

```
// 入侵方法A：最多資料個數
public static int hacking_data_A(ArrayList<int[]> original_domain, int[] hacker) {
    /*
     * @param original domain: 駭客入侵前的原始網域
     * @hacker: 駭客電腦資料
     * @return: 駭到的最多資料數 count
     */
    // TODO
}

// 入侵方法B：最高資料總價值
public static int hacking_data_B(ArrayList<int[]> original_domain, int[] hacker) {
    /*
     * @param original domain: 駭客入侵前的原始網域
     * @hacker: 駭客電腦資料
     * @return: 駭到的最高資料總價值 value
     */
    // TODO
}

// 印出結果
public static void printDomain() {
    /*
     * @return: 印出整個網域，數值與數值之間以空格區隔
     */
}
```

- `public static int[] hacking_data_A(ArrayList<int[]> original_domain, int[] hacker)`  
找到駭客電腦插入網域 `original_domain` 中可以駭到的最多資料數量，得到的最多資料個數 `count`。
- `public static int[] hacking_data_B(ArrayList<int[]> original_domain, int[] hacker)`

找到駭客電腦插入網域 `original_domain` 中可以駭到的最高資料價值，回傳得到的最高資料價值 `value`。

- `public static void printDomain(ArrayList<int[]> domain)`  
印出駭客入侵後的結果，該網域將包含 `hacker[]`。

以上函式可以依照自己的計算需求修改參數名，亦可新增計算所需的傳入參數，但**不可移除**。**未按規定者將斟酌扣分**。

## 輸入說明

- 第一行輸入 `n`，代表每台電腦具有的資料數量
- 第二行輸入 `m`，代表一個網域中的電腦數量
- 第三行開始輸入一個大小為 `n` 的一維陣列 `hacker[]`，代表駭客的電腦資料
- 接下來輸入一個 `m * n` 大小的 `domain`，型別為 `ArrayList<int[]>`
- 最後輸入一個字元 `method`，代表駭客的入侵方法

其中：

- `1 <= n, m <= 100` 為正整數
- `-50 <= hacker[i], domain.get(i)[j] <= 50` 為整數
- `method = 'A'` 或 `method = 'B'`

## 輸出說明

依照 `method` 指定的入侵方法(A 或 B)，具有不同的輸出：

- `method = 'A'`，輸出：

`At most` 最多資料個數 `data`。

最後印出入侵後的 domain。

- method = 'B'，輸出：

At most 最高資料價值 value.

最後印出入侵後的 domain。

其中，駭客入侵後的 domain 長相必唯一(行進路線可能不唯一)。

## 範例

Sample Input 1	Sample Output 1
6 5 0 3 8 4 6 0 0 1 2 3 4 5 1 1 0 0 0 0 1 0 1 2 3 0 0 1 2 3 2 0 0 3 2 3 2 0 A	At most 20 count. 0 1 2 3 4 5 1 1 0 0 0 0 0 3 8 4 6 0 1 0 1 2 3 0 0 1 2 3 2 0 0 3 2 3 2 0
Sample Input 2	Sample Output 2
6 5 0 3 8 4 6 0 0 1 2 3 4 5 1 1 0 0 0 0 1 0 1 2 3 0 0 1 2 3 2 0 0 3 2 3 2 0 B	At most 60 value. 0 1 2 3 4 5 1 1 0 0 0 0 0 3 8 4 6 0 1 0 1 2 3 0 0 1 2 3 2 0 0 3 2 3 2 0

## 範例說明

Sample Output 1	
<p>count = 20</p> <pre> 0   1   2   3   4   5   end 1   1   0   0   0   0 0   3   8   4   6   0   start 1   0   1   2   3   0 0   1   2   3   2   0 0   3   2   3   2   0 </pre>	<p>count = 20</p> <pre> 0   1   2   3   4   5   end 1   1   0   0   0   0 0   3   8   4   6   0   start 1   0   1   2   3   0 0   1   2   3   2   0 0   3   2   3   2   0 </pre>
Sample Output 2	
<p>value = 60</p> <pre> 0   1   2   3   4   5   end 1   1   0   0   0   0 0   3   8   4   6   0   start 1   0   1   2   3   0 0   1   2   3   2   0 0   3   2   3   2   0 </pre>	

## 繳交格式

### 1. 上傳內容須為 java 檔

- 程式碼(請繳交.java 檔)，不需要壓縮。
- 檔名為 AX\_學號(EX : A03\_112000000)

2. 程式碼開頭請標示

```
/*
```

```
Assignment 03
```

```
Name: 你的姓名
```

```
Student Number: 你的學號
```

```
Course 2024-課程代碼-你的班級
```

```
*/
```

3. 檔案編碼請統一使用：**utf-8**

4. 請針對程式內的重要功能進行「適當註解」，簡單說明該段程式碼的作用，註解的部分會納入評分考量，請勿隨意註解

5. 統一縮排格式，鋸齒縮排列入扣分項目。

以上繳交方式、內容錯誤的斟酌扣分，完整作業格式講說請見新 **ee-class** 公告中的附檔「作業格式及繳交說明.pdf」。若對作業還有問題，請寄信到助教信箱詢問：[ncuics112@gmail.com](mailto:ncuics112@gmail.com)。