

LeetCode 287.

Find the Duplicate Number

姓名: 林品安
學號: 1092923

目錄

- 題意說明
- 自己的思路 (二元搜尋、暴力解)
- 觀摩與學習 (雙指針解法)
- Floyd's Algorithm 解說
- Floyd's Algorithm 程式碼
- 總結

題意說明

- 給定一個陣列 `nums`，裡面有 `n+1` 個整數，而每個整數介於 $[1, n]$ 之間(包含)
- 證明陣列裡會有至少一組重複的數字 (題目給定陣列裡只會有一組重複的數字)
- 題目限制：不能改變 `array` 的順序，空間複雜度只能為 $O(1)$
- 題日期望：時間複雜度要為線性的
- 題目分類：雙指針(Two-Pointers)、二元搜尋(Binary-search)

範例1:

- 輸入：`[1,3,4,2,2]`
- 輸出：`2`

範例2:

- 輸入：`[3,1,3,4,2]`
- 輸出：`3`

自己的思路

暴力解 $O(n^2)$ (不符合題目限制)

- 先把陣列 **sort**，再遍歷整個陣列，當前後有兩個值一樣就回傳那個值

二元搜尋 $O(n \log n)$

- 每算一次 **mid** 值，就計算陣列裡面共有幾個數值小於等於 **mid**，將結果存於一個變數 **cnt**。如果 **cnt** 大於 **mid** 代表重複的數字會在 **mid** 的左邊(**mid** 也算是左邊範圍內)，反之則在右邊

```
1  class Solution {
2  public:
3      // binary-search
4      int findDuplicate(vector<int>& nums) {
5          int left=0, right=nums.size()-1, mid, cnt=0;
6          while(left < right){
7              cnt=0;
8              mid = (left+right)/2;
9              for(int i=0; i<nums.size(); i++){ // 算出有幾個數字小於等於 mid
10                 if(nums[i] <= mid)
11                     cnt++;
12             }
13             if(mid < cnt)    right = mid; // 代表重複的數會在 mid 左邊
14             else    left = mid+1;
15         }
16         return right;
17     }
18 };
```

範例 I:

- 輸入 : [1,3,4,2,2]
- 輸出 : 2

<Sol>

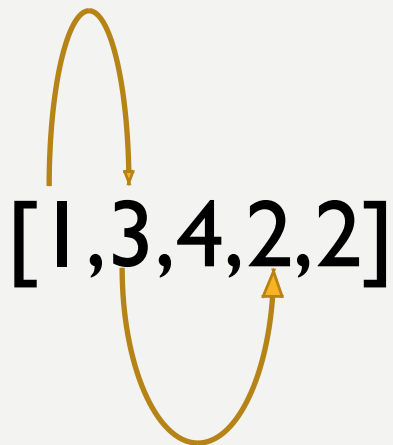
1 st	2 nd	3 rd
mid = 2	mid = 1	return right
cnt = 3	cnt = 1	
right = 2	left = 2	

觀摩與學習

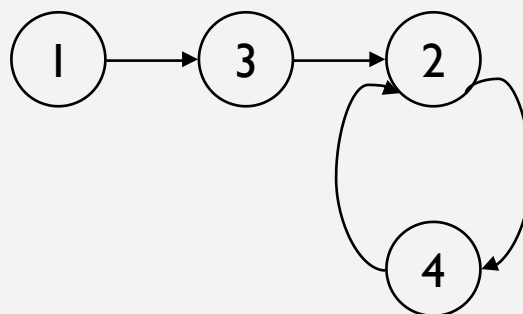
- 題目要求時間複雜度 $O(n)$ 且空間複雜度 $O(1)$ 又不能修改到陣列
- 改用 Floyd's Algorithm 來解，又稱為龜兔賽跑演算法

範例1 : [1,3,4,2,2]，把題目換成 linked list 把演算法視覺化

Step1: 把每個陣列值想成指標



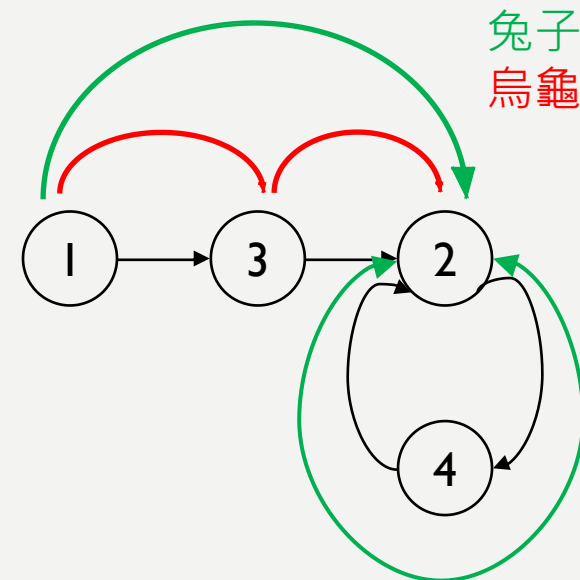
Step2: 轉換成 linked-list



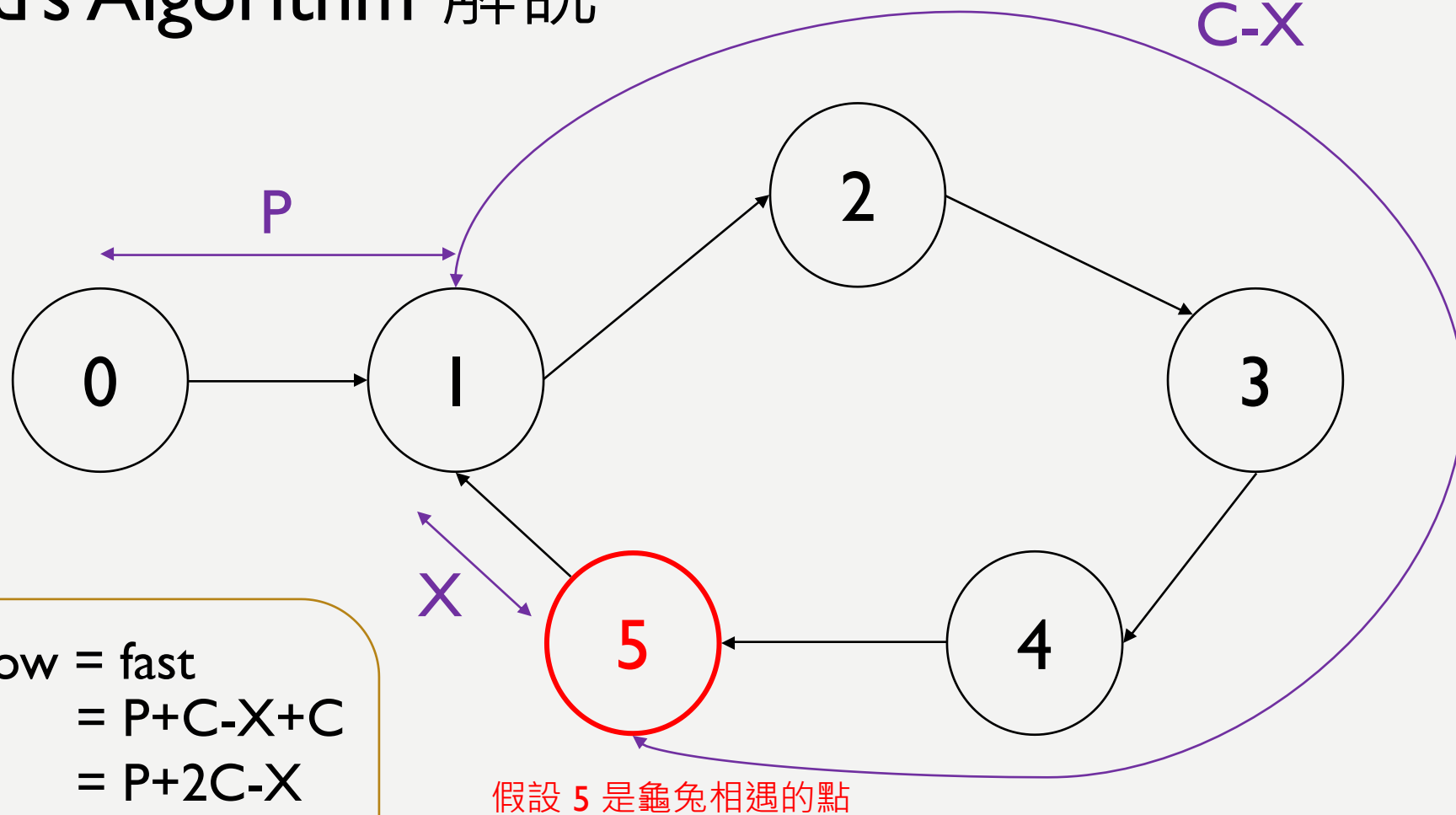
Step3: 利用 Floyd's Algo. 找到循環起點

龜兔從起點同時出發，龜走一步，兔走兩步。

當龜兔相遇，龜退回起點，兔原地待命，龜兔同時出發，龜走一步，兔也走一步。龜兔相遇之處即是循環起點。



Floyd's Algorithm 解說



$$\begin{aligned} 2 \times \text{slow} &= \text{fast} \\ &= P + C - X + C \\ &= P + 2C - X \end{aligned}$$

$$\rightarrow 2(P + C - X) = P + 2C - X$$

$$P = X$$

- 令 m 是出發起點到循環起點的距離， n 是循環長度。烏龜最多走 $m + n$ 步，兔最多走 $2m + 2n$ 步，時間複雜度 $3m + 3n = O(m + n)$

```

1  class Solution {
2  public:
3      // Floyd's Algo
4      int findDuplicate(vector<int>& nums) {
5          int hare=0, tortoise=0;           // 兔子與烏龜都從起點出發
6          while(true){
7              hare = nums[nums[hare]];       // 兔子每次都走兩步
8              tortoise = nums[tortoise];    // 烏龜每次都走一步
9              if(hare == tortoise)          // 相遇的時候
10                 break;
11          }
12          hare = tortoise; tortoise=0;      // 兔子原地待命，烏龜回到起點
13          while(true){
14              tortoise = nums[tortoise];    // 烏龜每次走一步
15              hare = nums[hare];            // 兔子每次也只走一步
16              if(tortoise == hare)          // 相遇的時候，代表為循環的起點
17                 break;
18          }
19          return tortoise;
20      }
21 };

```

總結

- 雙指針、二元搜尋的題目
- 以 Floyd's Algorithm 可以在線性時間解決題目
- Floyd's Algorithm 是一個可以找到 linked-list 裡 cycle 起點與 cycle 長度的有效率方式
- 這題是很多大公司面試出的題目 (e.g. Google, Microsoft, Amazon, Facebook)