

Problem definition:

給定一由 n 個整數組成的已排序陣列 $nums$,

而假設此陣列會做若干次循環位移

我們欲在循環位移後的陣列, 查找一元素 $target$

並回傳 $index$, 若無, 回傳 -1 . 且每個數皆 $unique$

其中: 循環位移的定義為設原 $nums$ 為 $nums[0], \dots, nums[n-1]$

在其中選定一 $pivot$ 為 p , $1 \leq p < n$

位移後的陣列為: $nums[p], \dots, nums[n-1], nums[0], \dots, nums[p-1]$

Algorithm:

(1). 在 $nums$ 中找出斷點位置

(2). 知道 $pivot$ 位置後, 可對 $nums$ 做 Binary Search 找 $target$.

Key Points:

(1). 找斷點位置需為 $O(\lg n)$ 時間 \Rightarrow Binary Search

(2). 要做 $index$ shifting 才能對已知 $pivot$ 的 $nums$ 做 BS_{th}

Example: $num = [1\ 2\ 3\ 4\ 5\ 7\ 0]$, $target = 2$

原 $num = [0\ 1\ 2\ 3\ 4\ 5\ 7]$, $pivot = 1$

如何設計 Binary Search 來找斷點為何。

1.
 \Rightarrow Pivot 性質: 在原 num 中, $\forall i, i < pivot, num[i] < num[pivot]$
 $i > pivot, num[i] > num[pivot]$

2.
在 num 中, $num[0] = \text{原 } num[pivot]$

斷點定義: 設原 $num[0]$ 在 num 的 index 為斷點 index.

設為 b

則 $num[b] < num[0]$

$num[b-1] \geq num[0]$

Example: case I.
 $num = [1\ 2\ 3\ 4\ 5\ 7\ 0]$

$b = 6, num[6] = 0 < num[0] = 1$

$num[5] = 7 > num[0] = 1$

case II.

$$\text{nums} = [7, 0, 1, 2, 3, 4, 5]$$

$$b = 1, \quad \text{nums}[b] = 0 < 7$$

$$\text{nums}[b] = 7 = 7$$

⇒ 找到 mid 和 斷點關係，如何決定 prune 掉左半/右半子陣列。

case I.

$$\text{nums} = [1, 2, 3, 4, 5, 7, 0]$$

$$\text{nums}[\text{mid}] = 4 \quad \text{nums}[\text{mid}-1] = 3$$

$$\therefore 4 > 1 \quad \underline{\text{且}} \quad 3 > 1$$

∴ b 肯定在右半子陣列。

case II.

$$\text{nums} = [7, 0, 1, 2, 3, 4, 5]$$

$$\text{nums}[\text{mid}] = 2 \quad \text{nums}[\text{mid}-1] = 1$$

$$\therefore 2 < 7 \quad \underline{\text{且}} \quad 1 < 7$$

$\therefore b$ 肯定在左半子陣列中.

Case III.

$numr = [9 \ 12 \ 3 \ 4 \ 5 \ 7]$

斷點定義失敗, \therefore 不連續.

但我們可以判斷 $numr[0] < numr[n-1]$

來 check 此 edge case.

Algorithm: (1). 比較陣列中間元素, 並可將原陣列切成左半右半子陣列。

(2). 若 $numr[0] < numr[n-1]$, 則 $pivot = 0$,

(3). 若 $numr[mid] < numr[0]$

且 $numr[mid-1] \geq numr[0]$

則 mid 為 b

(4). 若 $numr[mid] > numr[0]$

且 $numr[mid-1] > numr[0]$

則 b 肯定在右半子陣列.

5). 若 $nums[mid] < nums[0]$

且 $nums[mid-1] < nums[0]$

則 b 肯定在左半子陣列.

(6). 持續在子陣列尋找, 直至找到停止.

Example:

1.

[1 2 3 4 5 7 0]



[5 7 0]



[0] \Rightarrow Base case

2.

[7 0 1 2 3 4 5]



[7 0 1] \Rightarrow Base case.

3. $[3457012]$



$[012]$



$[0] \Rightarrow \text{Base case}$

找到斷點後, 如何利用斷點來對 num 做 Binary Search.

用 example 來看: $\text{nums} = [3457012]$

已知斷點 index 為 4

我們想用 $[0123457]$ 來做 BS

未 shifting 前 index: 0123456

↓ ↓ ↓ ↓ ↓ ↓ ↓

shifting 後 index: 4560123

可直接用: $(\text{index} + b) \% n$ 來做 index shifting

在 num 取值.

e.g. 0 1 2 3 4 5 7

$$\text{left} = 0, \text{right} = n-1 = 6$$

$$\text{mid} = \left\lfloor \frac{0+6}{2} \right\rfloor = 3$$

等價於：

3 4 5 7 0 1 2

$$\begin{aligned} \text{left} &= (0+4) \% 7 & \text{right} &= (6+4) \% 7 \\ &= 4 & &= 3 \end{aligned}$$

$$\begin{aligned} \text{mid} &= (3+4) \% 7 \\ &= 0 \end{aligned}$$

另一个找断点的 Algorithm:

1. $left = 0, right = n - 1$
2. while ($left \leq right$)
3. $mid = (left + right) / 2$
4. if ($nums[mid] > nums[n-1]$) $left = mid + 1$
5. else $right = mid - 1$
6. return $right + 1$

Example: $nums = [3, 4, 5, 7, 0, 1, 2]$

pass 1. $nums[mid] = 7 > 2 \therefore left = 4$

$\Rightarrow [0, 1, 2]$

pass 2. $nums[mid] = 1 < 2 \therefore right = 4$

$\Rightarrow [0]$

pass 3: $nums[mid] = 0 < 2 \therefore right = 3$

$\Rightarrow []$

return 4 _{#.}

Ex 2. $nums = [0, 1, 2, 3, 4, 5, 7]$

pass 1. $nums[mid] = 3 < 7 \quad \therefore right = 2$

$\Rightarrow [0, 2]$

pass 2. $nums[mid] = 1 < 7 \quad \therefore right = 0$

$\Rightarrow [0]$

pass 3. $nums[mid] = 0 < 7 \quad \therefore right = -1$

$\Rightarrow []$

return 0

Why the algorithm will work?

再次考慮斷點性質:

$nums = [3, 4, 5, 7, 0, 1, 2]$

斷點為 0, 可以發現, 在斷點左方的數皆大於 $nums[n-1]$
而右方的數皆小於 $nums[n-1]$

而斷點為大於 $num[n-1]$ 的第一個數。

∴ 比較 mid 和 $num[n-1]$ 時，

若 $mid > num[n-1]$ ，則必往右半走

$mid < num[n-1]$ ，則必往左半走。

又因斷點自己 $< num[n-1]$ ，必定是 $right$ 走到斷點 $index - 1$ 停止。

Variation: 若 num 中的元素不為 $unique$ ，會如何影響

Algorithm 正確性？

Example, $nums\ 1 = [2\ 2\ 2\ 2\ 3\ 2\ 2]$

$nums\ 2 = [2\ 3\ 2\ 2\ 2\ 2\ 2]$

若 $mid == nums[n-1]$ ，不能確保斷點在左還右

不能決定要向左還向右，皆得考慮！