

Binary Search (二元搜尋法)

- 目標：

在已經排序好、長度為 n 的陣列裡，

1. 尋找給定的元素 k
2. 判斷給定的元素 k 是否在陣列裡

- 所需計算量 (時間複雜度)： $O(\log n)$

Binary Search (二元搜尋法)

```
function binary_search(A, n, T):  
    L := 0  
    R := n - 1  
    while L <= R:  
        m := floor((L + R) / 2)  
        if A[m] < T:  
            L := m + 1  
        else if A[m] > T:  
            R := m - 1  
        else:  
            return m  
    return unsuccessful
```

Binary Search (二元搜尋法)

- 想法：

既然陣列 A 已經排序好了，那麼把待搜尋的範圍裡，中間的元素拿出來，跟 k 比比看

– 若不相等，則可以把範圍縮小為原本的一半

- 每個回合，都可以丟掉一半的元素。

Binary Search (二元搜尋法)

- 事實上，適用 Binary Search の場合，不只是排序好的陣列。
- 在單調 (monotone) 的 $\{0, 1\}$ -序列中，尋找 最後一個 / 第一個 0 / 1

Binary Search (General Form)

0, 0,, 0, 1, 1, 1,, 1

尋找序列裡，第一個 1 的位置

- 使用變數：
 - left 永遠指向 0, right 永遠指向 1
 - 當範圍裡只剩兩個元素時，搜尋結束
 - 複雜度 $O(\log(\text{Range})) * \text{Time}(\text{Per Test})$