

二分搜尋法



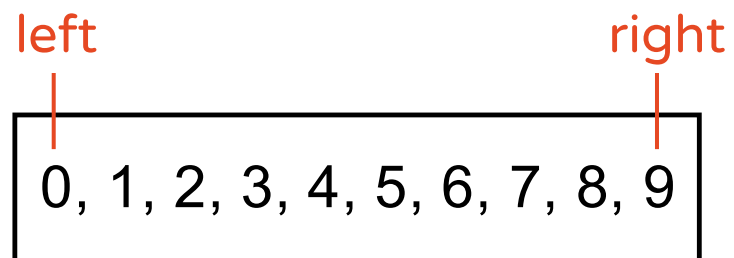
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

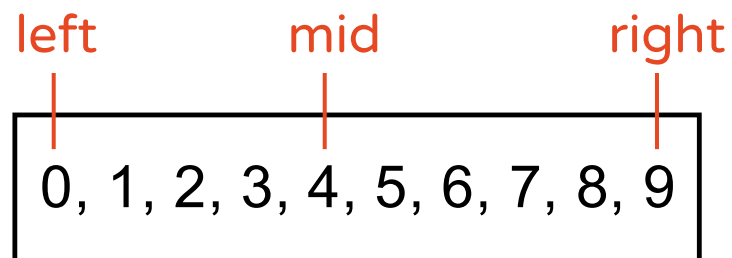
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



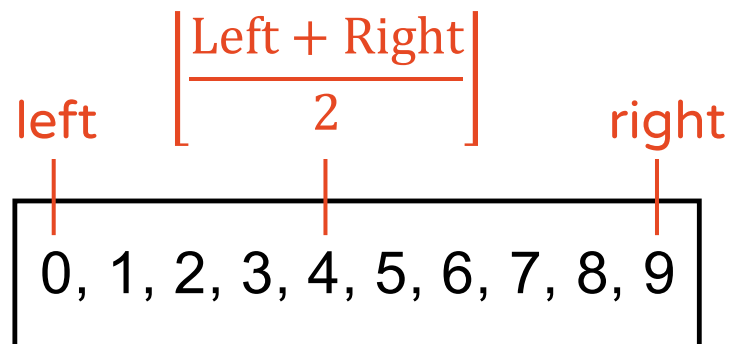
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



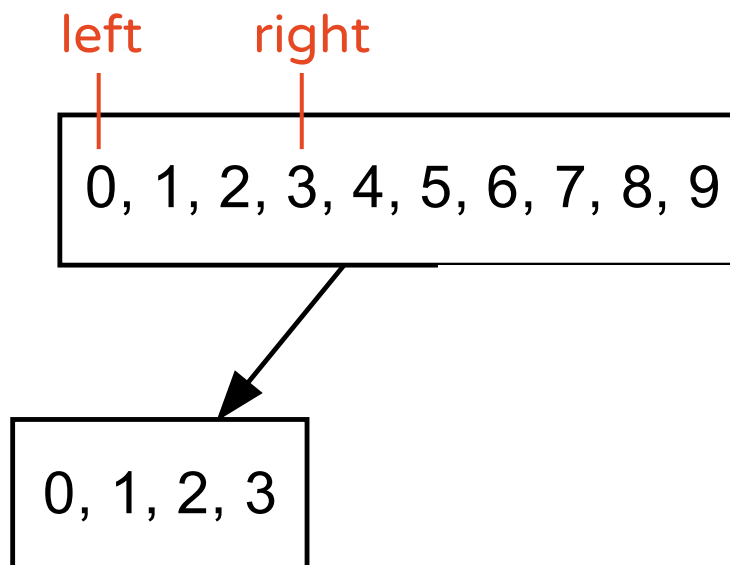
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



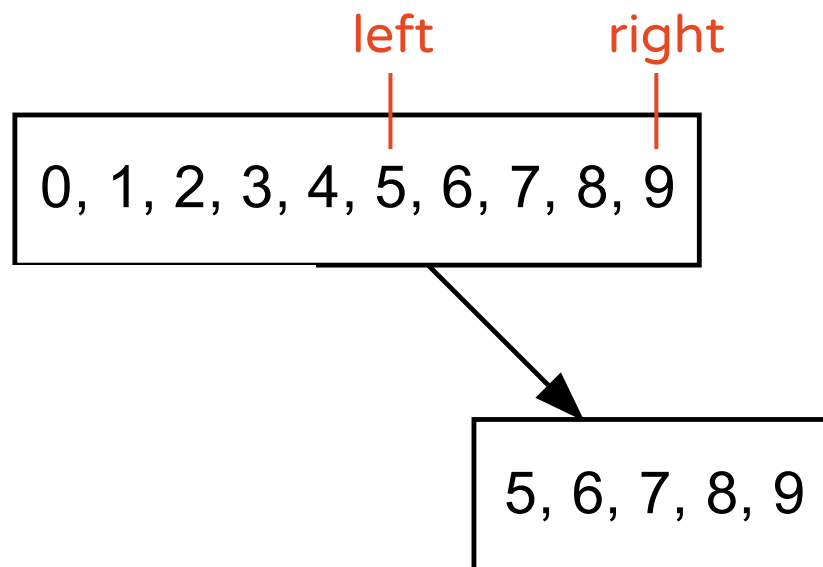
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



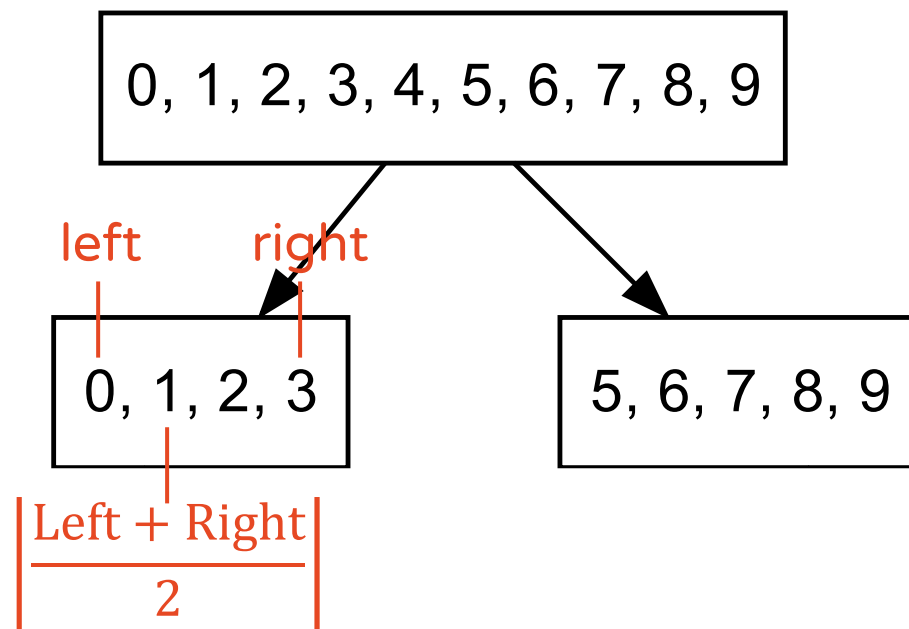
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



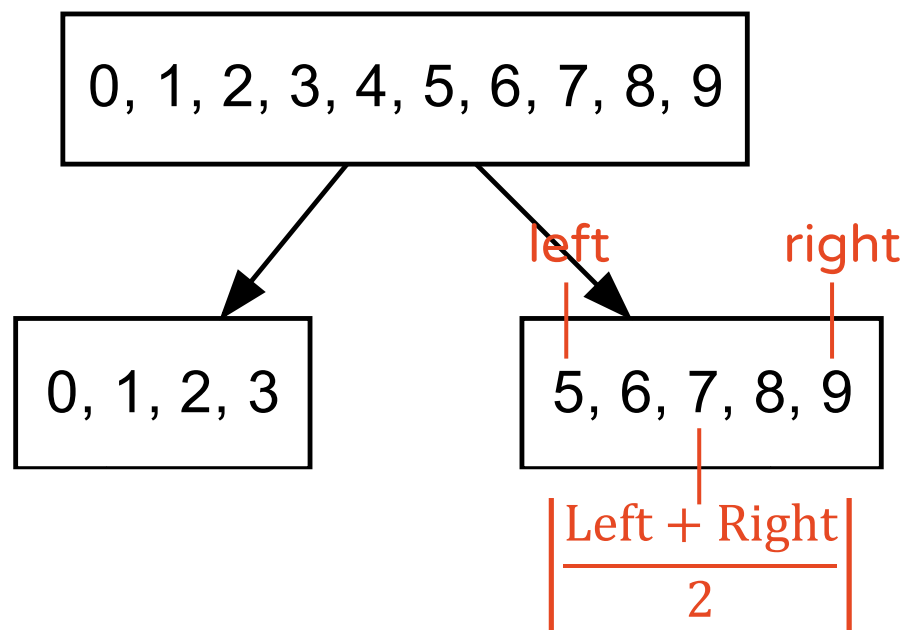
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



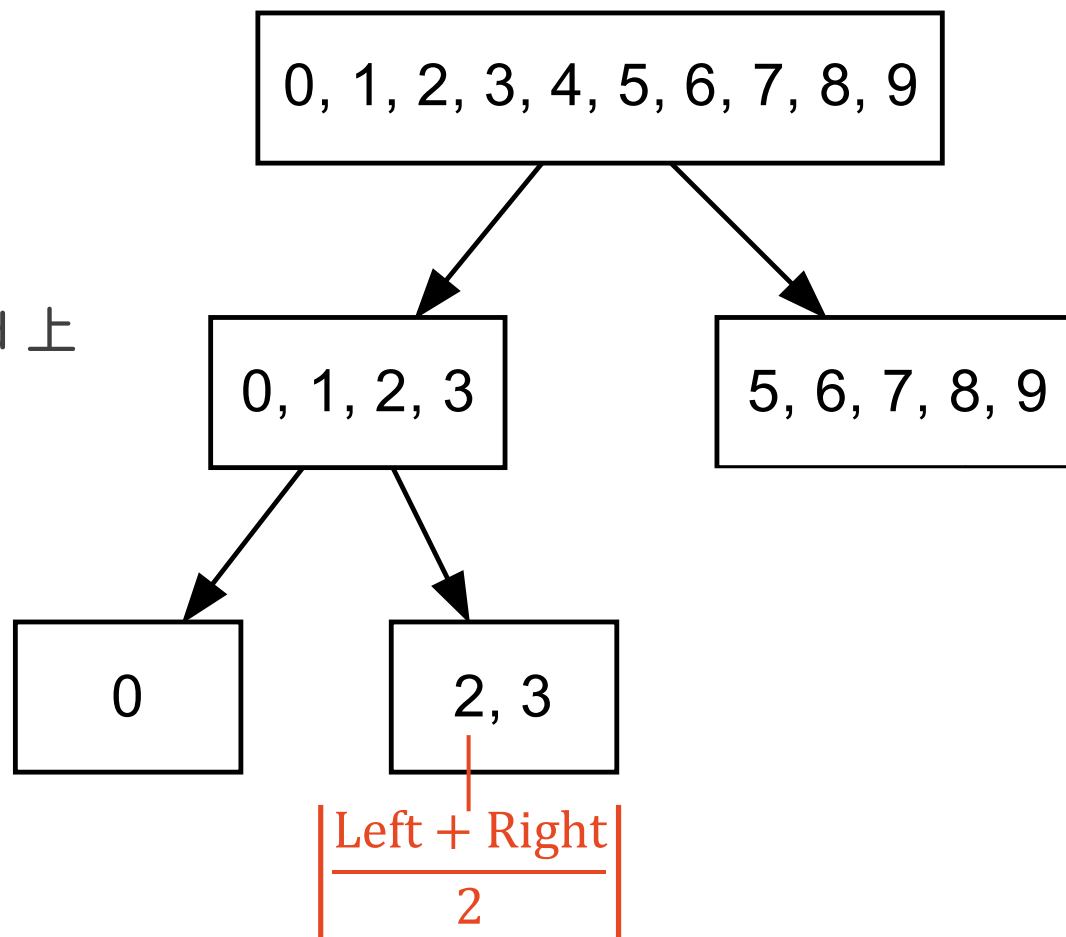
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



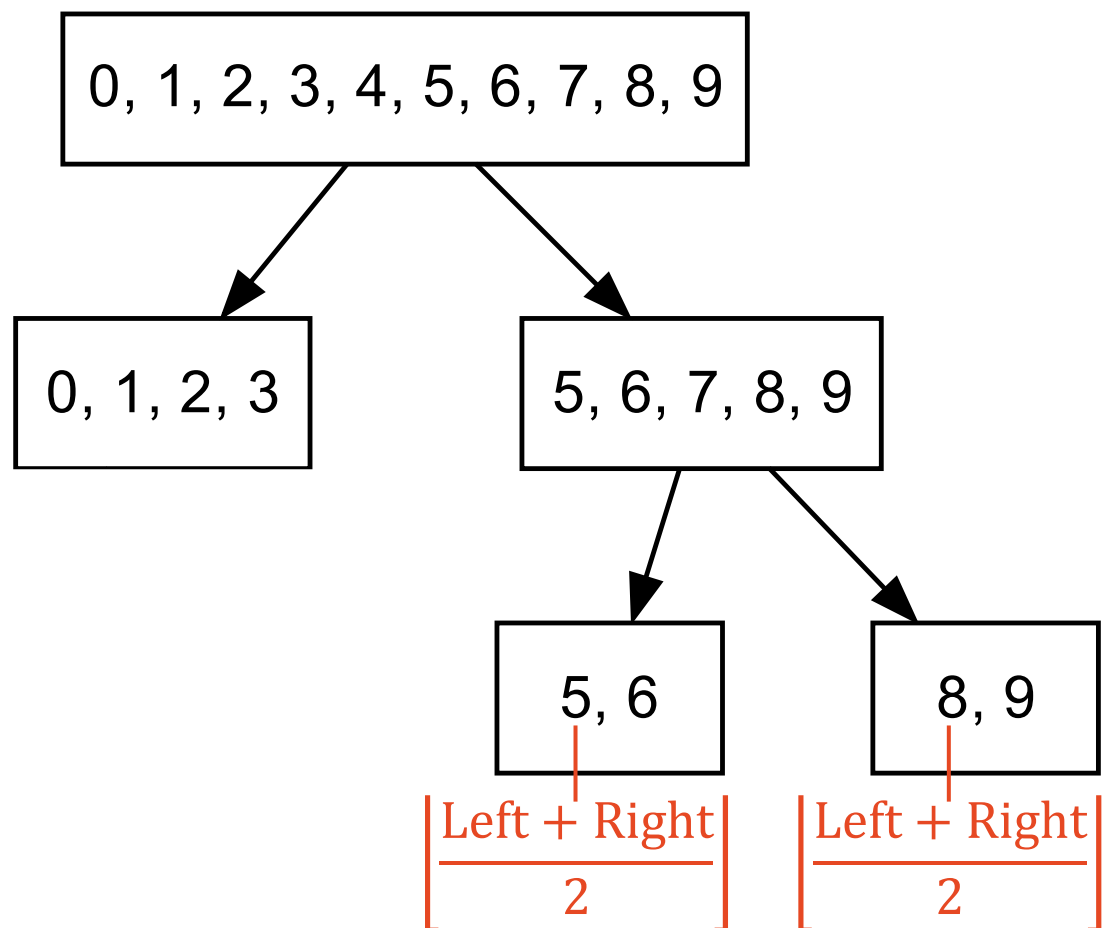
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



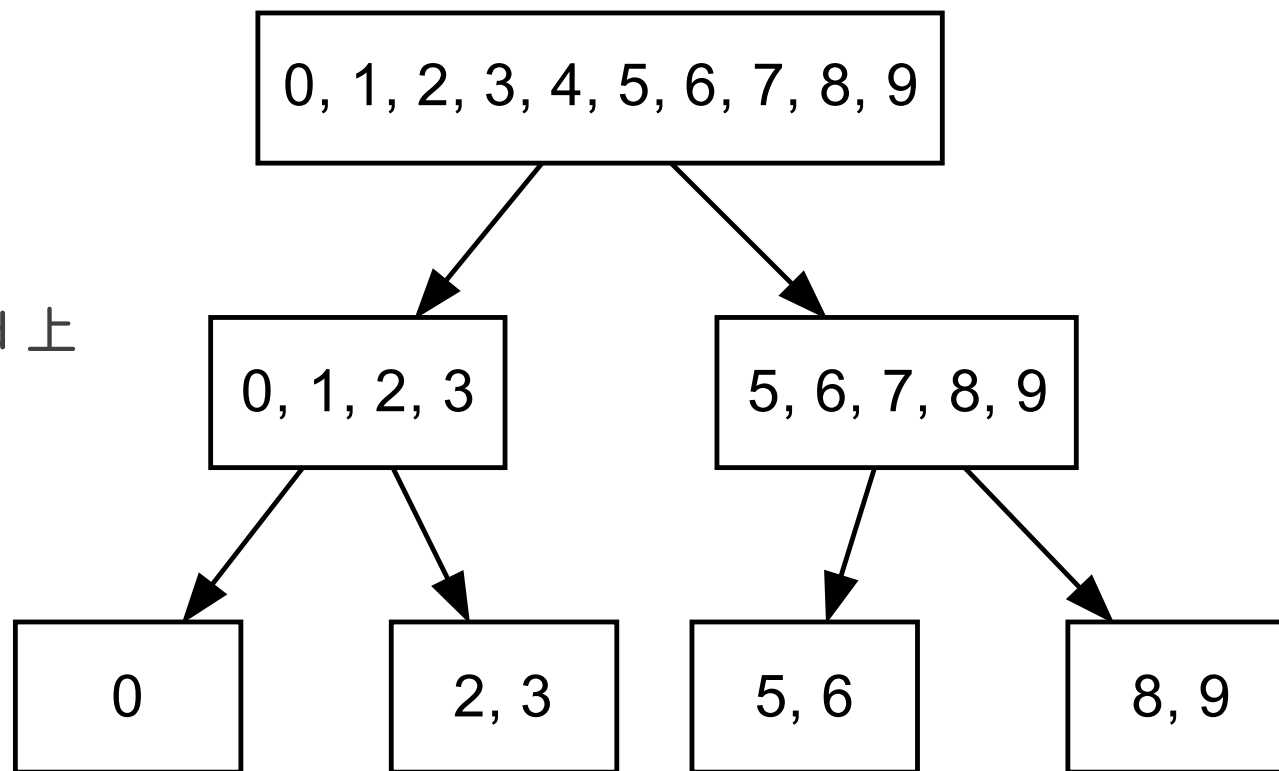
「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



「持續平分陣列，直到指針碰到搜尋目標」

1. 給定一個陣列從 0 到 9
2. 找出每個節點的指針
3. 如果中間元素是目標就結束
4. 否則判斷大小，移動指針到 Mid 上



Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

- 輸入：
 - 第一行輸入一個整數 n 為元素數量
 - 第二行有 n 個整數 $x_1, x_2 \cdots x_n$
- 輸出：寫出一個整數代表序列中有幾個不重複的數字

輸 入

5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

思路 1

- 持續尋找是否存在 $k+1$ 這個數字
- 如果有就設定計數器+1，否則不變
- 當找到 $k + 1$ 即代表不重複的數字

輸 入

5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

2	3	2	2	3	
---	---	---	---	---	--

陣列初始化

上限值：-1

單獨數：0

輸 入

5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

2	2	2	3	3	
---	---	---	---	---	--

排序陣列

上限值：-1

單獨數：0

輸入

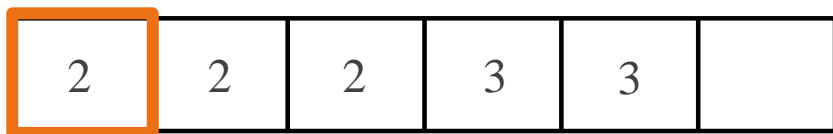
5
2 3 2 2 3

輸出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數



開始遍歷，找到第一個單獨數，更新上限值與單獨數

上限值：**2**

單獨數：**1**

輸 入

5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

2	2	2	3	3	
---	---	---	---	---	--

與上限值相同，skipping

上限值：2

單獨數：1

輸入

5
2 3 2 2 3

輸出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

2	2	2	3	3	
---	---	---	---	---	--

上限值：2

單獨數：1

輸 入

5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

2	2	2	3	3	
---	---	---	---	---	--

找到第二個單獨數，更新上限值與單獨數

上限值：**3**

單獨數：**2**

輸 入

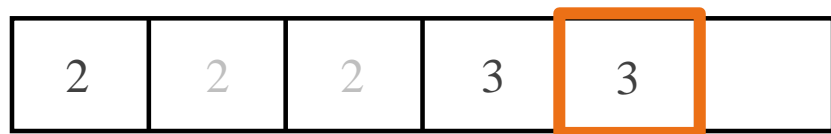
5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數



與上限值相同，skipping

上限值：3

單獨數：2

輸入

5
2 3 2 2 3

輸出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數



上限值：3

單獨數：2

輸入

5
2 3 2 2 3

輸出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

2	2	2	3	3	EOF
---	---	---	---	---	-----

輸出**單獨數**數，完成題目

上限值：3

單獨數：2

輸 入

5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

思路 2

- 把重複的數字直接去除
- 用 `std::set()`

輸 入

5
2 3 2 2 3

輸 出

2

Distinct Numbers

給定一個包含 n 個整數的清單，您的任務是計算清單中不同值的個數

```
#include <iostream>
#include <set>

using namespace std;

int main() {
    int n;
    cin >> n;

    int arr[n];

    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    set<int> result(arr, arr + n);
    cout << result.size() << '\n';
}
```

輸入

5
2 3 2 2 3

輸出

2

STD 工具

不用自己手刻二分搜尋函數

upper_bound

- 在數列中 「大於目標的最小值的索引」
- 搜尋數列中的目標值 5，得到第一個比 5 大的數字的位置索引

lower_bound

- 在數列中 「大於或等於目標的最小值的索引」
- 搜尋數列中的目標值 5，得到第一個大於或等於 5 的數字的位置索引。

STD 工具

不用自己手刻二分搜尋函數

```
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int target;
    cin >> target;

    int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

    auto u = upper_bound(arr, arr + size(arr), target);
    cout << (u - arr - 1) << endl;

    auto l = lower_bound(arr, arr + size(arr), target);
    cout << (l - arr) << endl;
}
```

問題

效率問題

「Distinct Numbers」

這個題目可透過 set 或 sort 來解決

假設面對的是連續的數字串流而非預先定義的陣列，
並且不能使用 `set.insert()` 的情況下

如何有效地即時計算不同數字的個數？

謝謝聆聽

