

Sort & Binary Search

4/26

排序的理由？

- 今天想找 A 班上的 3 號同學
- 全班正在玩大風吹，所以他不在第三個位置上
- 我們怎麼做才能找到他？

排序的理由？

- 今天想找 A 班上的 3 號同學
- 全班正在玩大風吹，所以他不在第三個位置上
- 我們怎麼做才能找到他？

一個一個問？

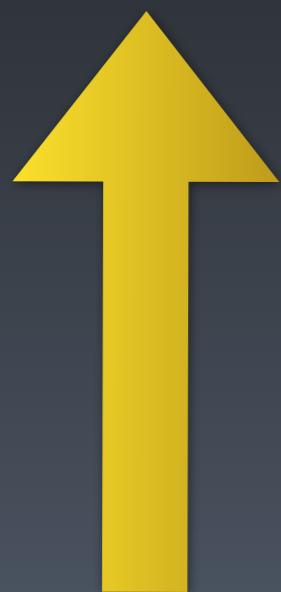
1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

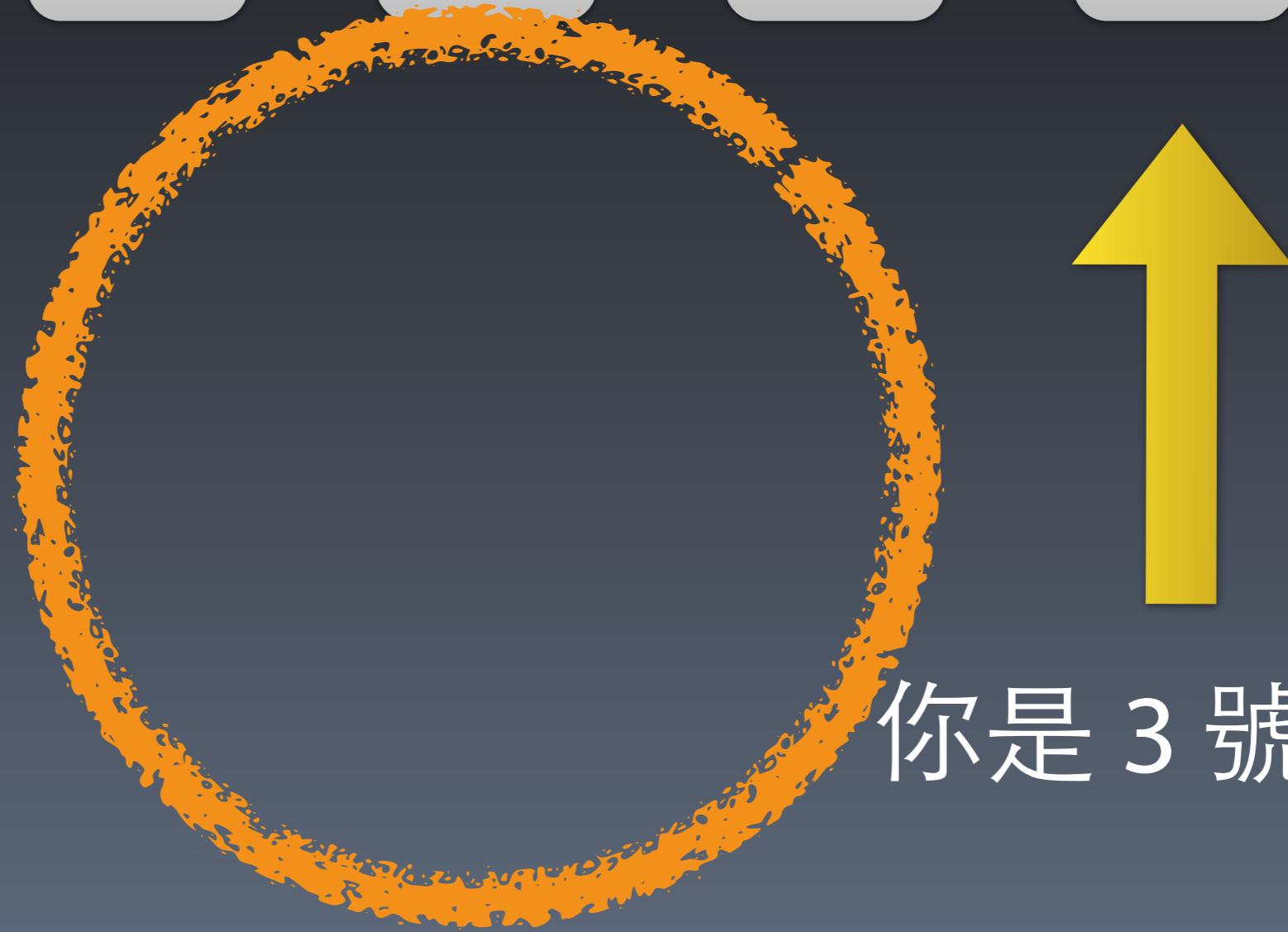
1

5

4

2

3



你是 3 號嗎？

老師表示：
給我排好！！！

1

2

3

4

5

1

2

3

4

5

3 號出來！

1

2

4

5

3

3 號出來！

bubble sort



21

53

42

9

13

21

53

42

9

13

21

53

42

9

13

53

21

42

9

13

53

21

42

9

13

53

21

42

9

13

53

42

21

9

13

53

42

21

9

13

53

42

21

9

13

53

42

21

9

13

53

42

21

9

13

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

53

42

21

13

9

bubble ?

- 從最左邊（或最右邊）開始
- 數字會慢慢像汽泡一樣的浮到對應的位置
- 再看一次？

執行路徑？？



程式碼

```
for (int i = 0; i < size; i++) {  
    for (int j = 0; j < size-1-i; j++) {  
        if (arr[j] < arr[j+1]) {  
            swap(arr[j], arr[j+1]);  
        }  
    }  
}
```

Counting Sort

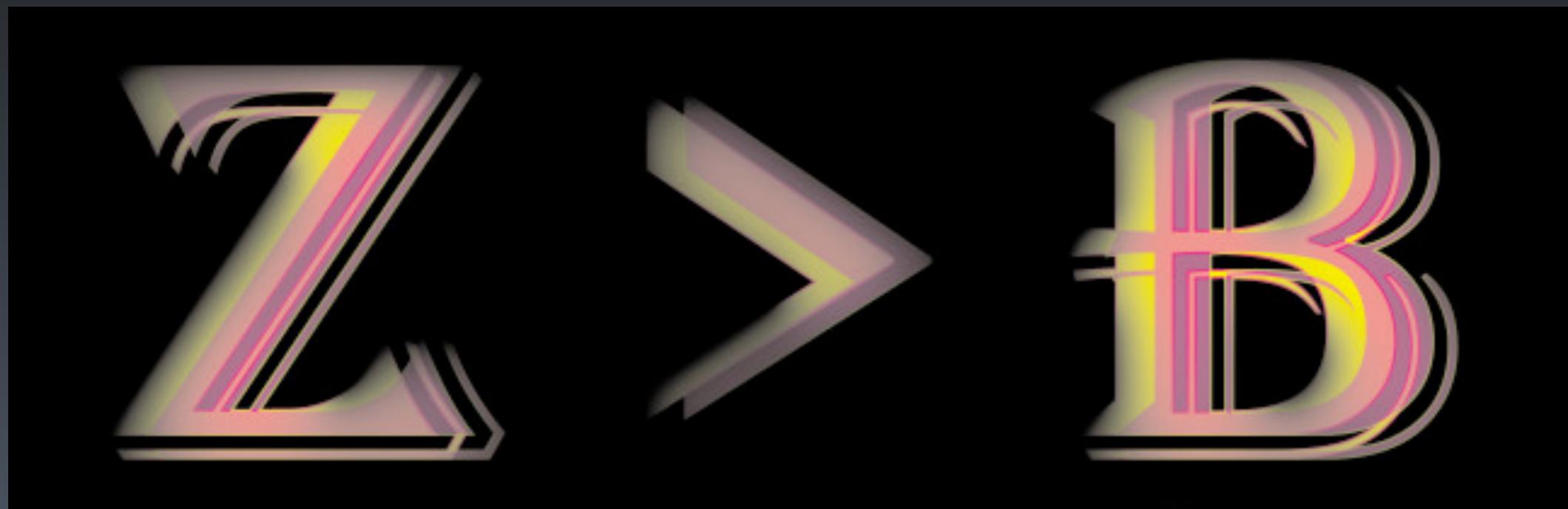
Counting Sort??

- 今天全校大會考，參與考試的人共有一萬
- 成績從 0~100
- 現在想問你前五多人得到的成績分別為何
- 老方法？

Counting Sort??

- 今天全校大會考，參與考試的人共有一萬
- 成績從 0~100
- 現在想問你前五多人得到的成績分別為何
- 老方法？

一個一個問？？？



10000 >> 100

莫忘箱子
注重簡化

0



0



0



0



0



1



1



2



3



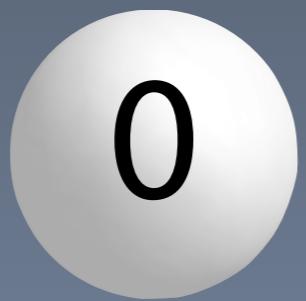
2



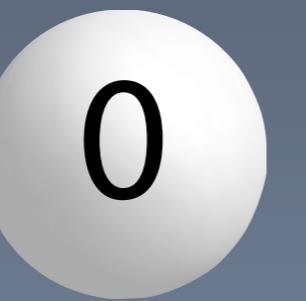
1



0



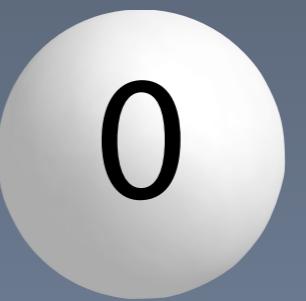
0



1



0



1



1



0



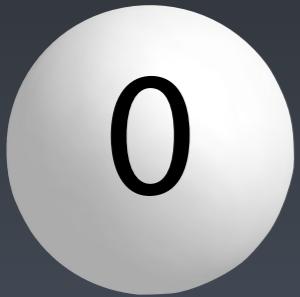
0



0



0



1



1



2



3



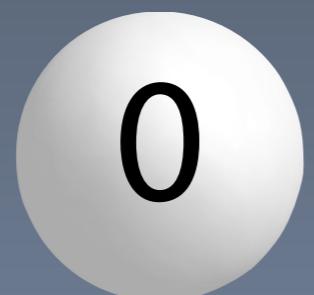
2



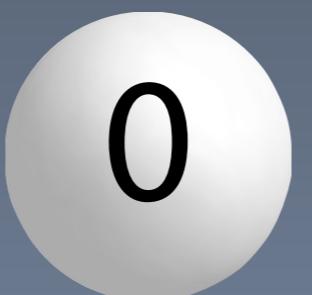
1



0



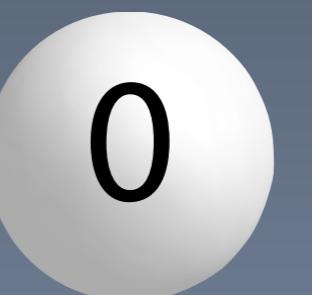
0



1



0



1



1



1



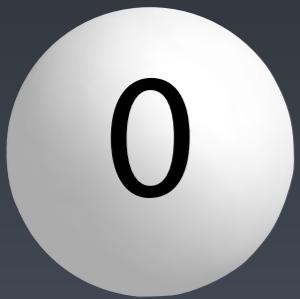
0



0



0



1



1



2



3



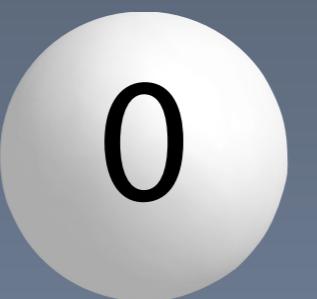
2



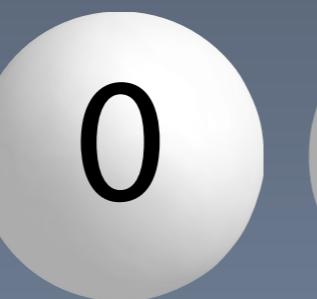
1



0



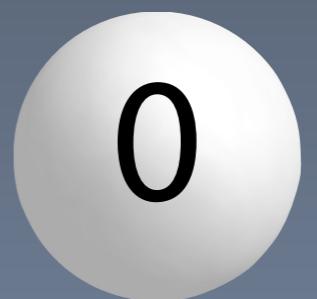
0



1



0



1



1



2



0



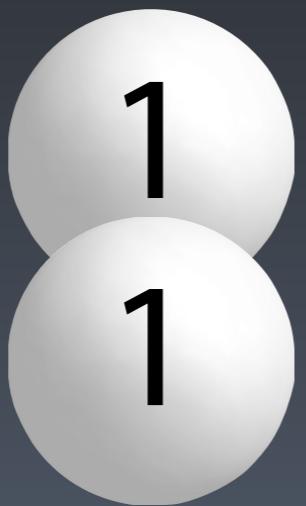
0



0



1



2



3



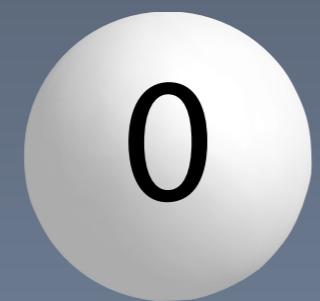
2



1



0



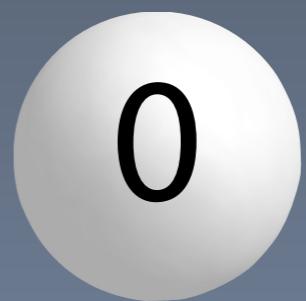
0



1



0



1



1



2



1



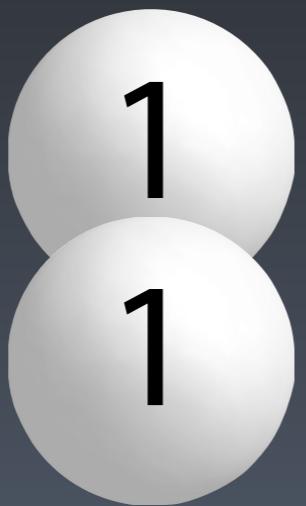
0



0



1



2



3



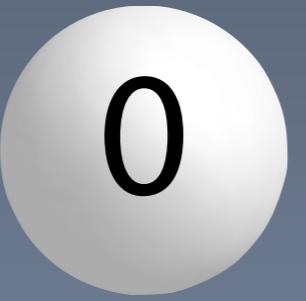
2



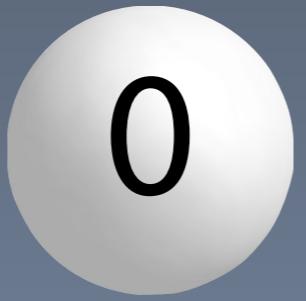
1



0



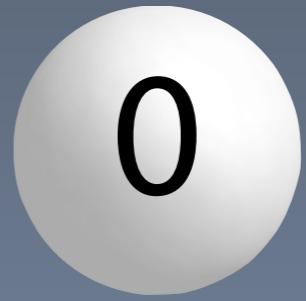
0



1



0



1



1



0

2



1

1



2

1

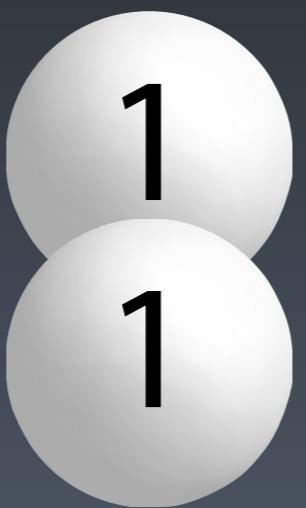


3

0



1



1

2



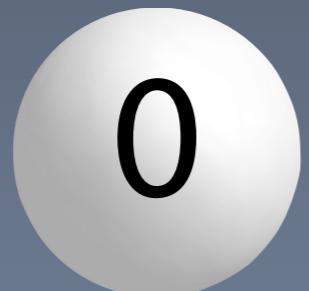
3



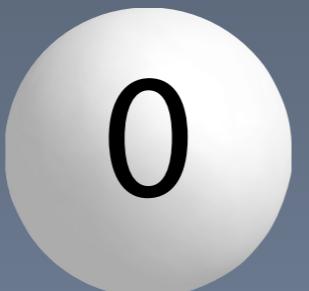
1



0



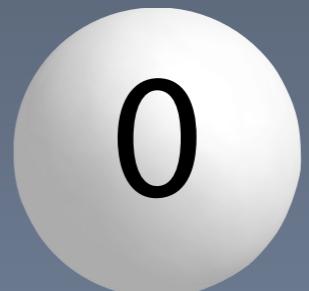
0



1



0



1



2



1



2



2



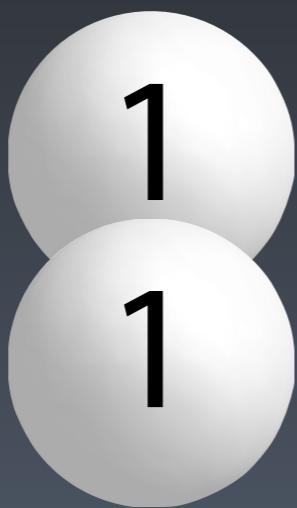
1



0



1



2



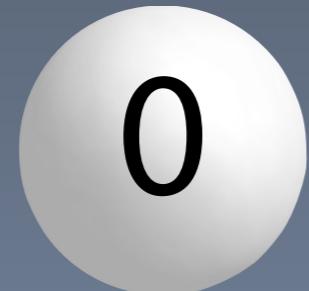
3



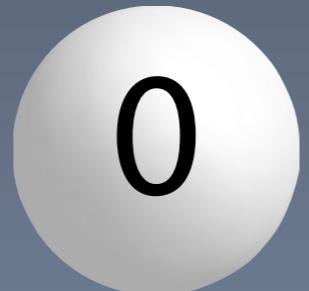
1



0



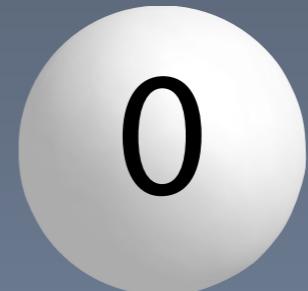
0



1



0



1



1



0

3



1

2



2

1



3

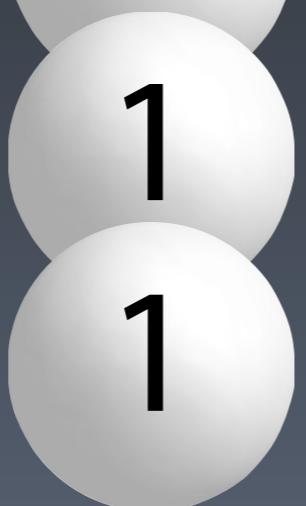
0



1



1



1

2



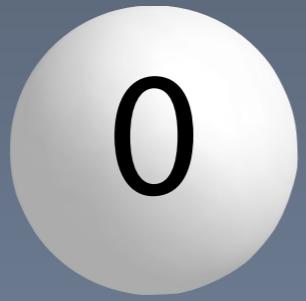
2



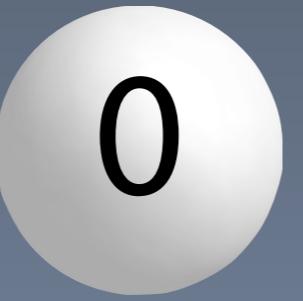
3



0



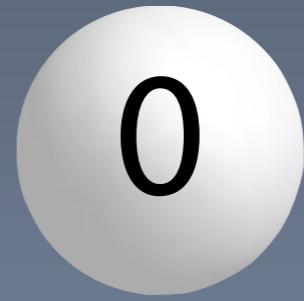
0



1



0



1



2



3



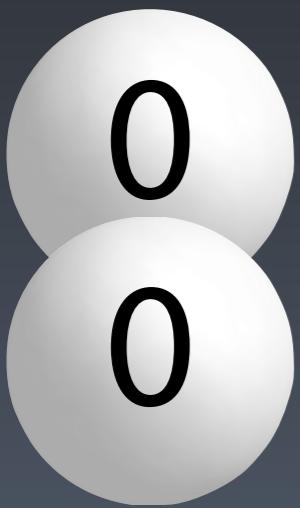
2



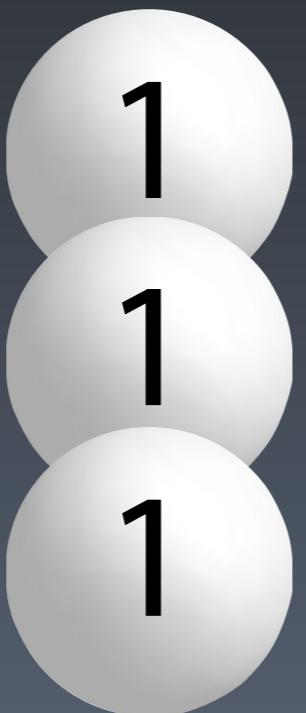
1



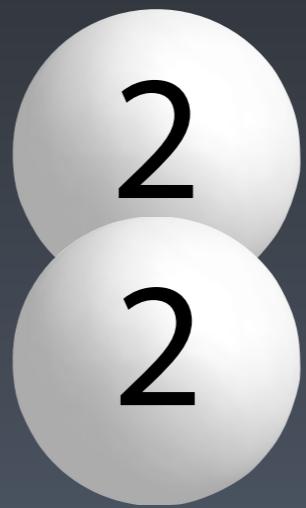
0



1



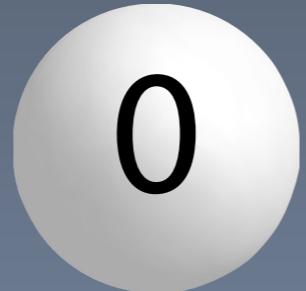
2



3



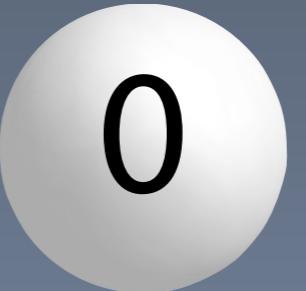
0



1



0



1



3



3



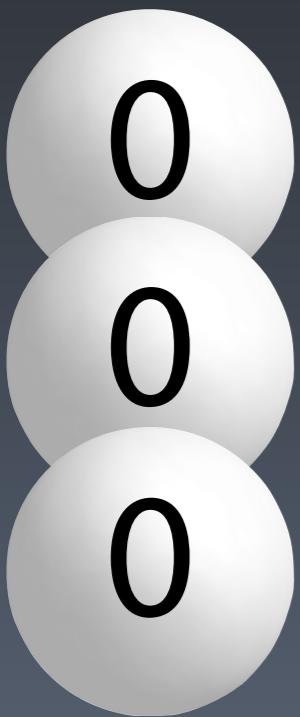
2



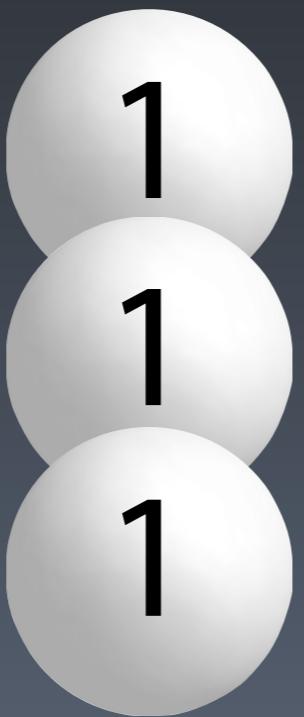
1



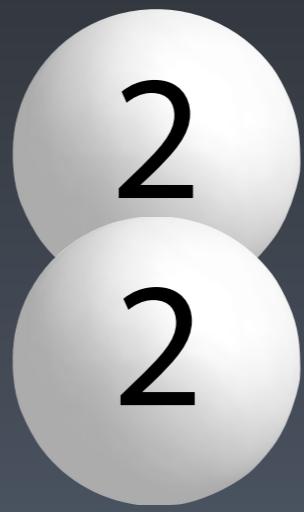
0



1



2



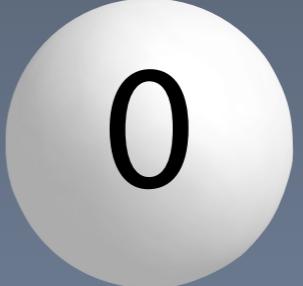
3



1



0



1



3



4



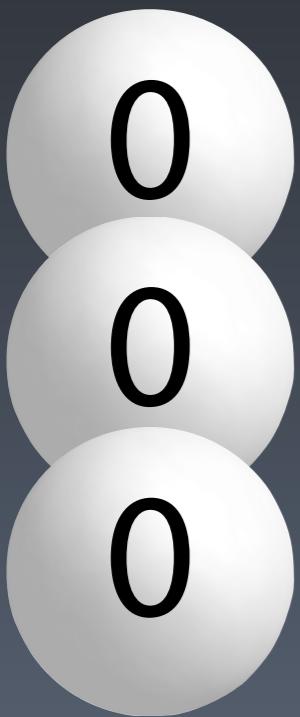
2



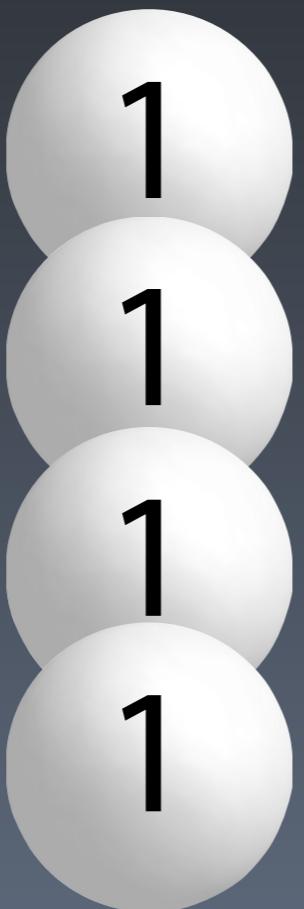
1



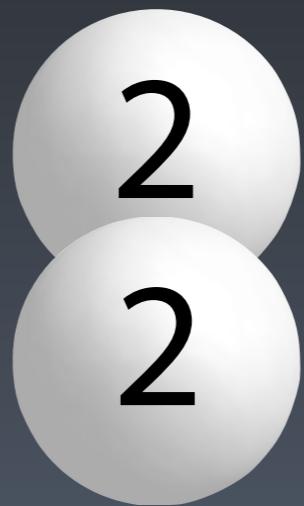
0



1



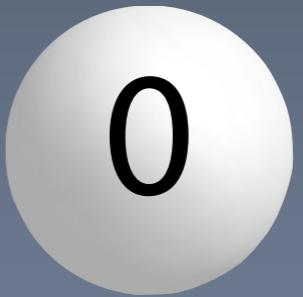
2



3



0



1



4



4



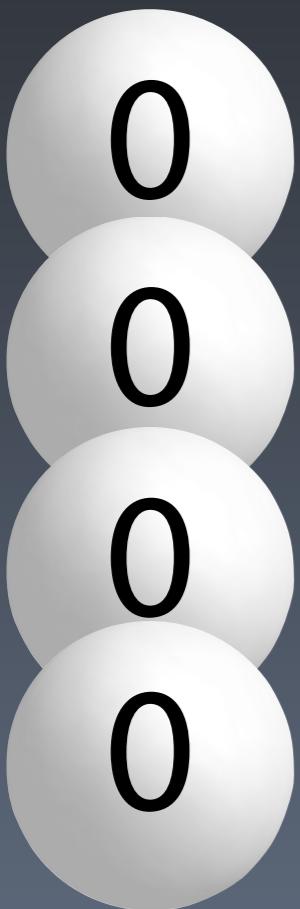
2



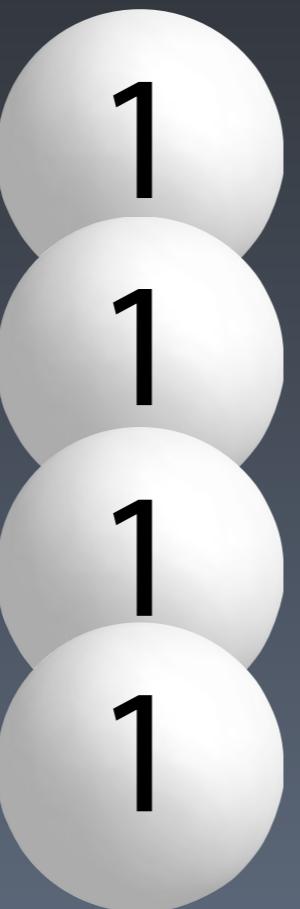
1



0



1



2



3



1



4



5



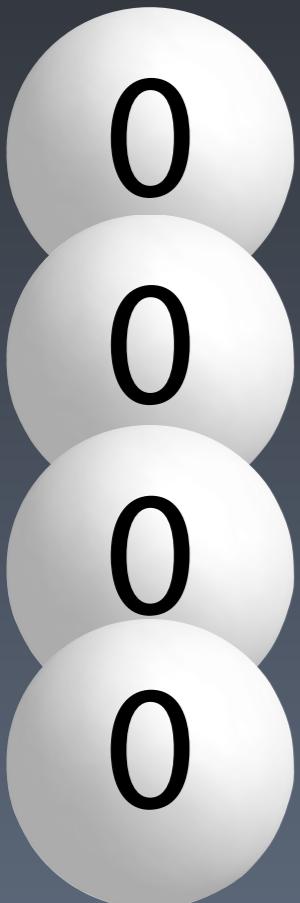
2



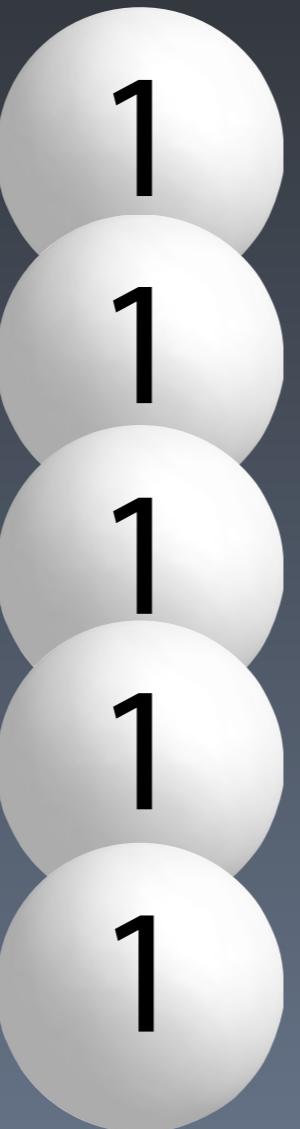
1



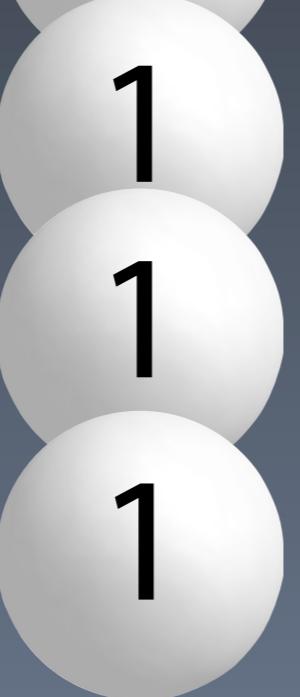
0



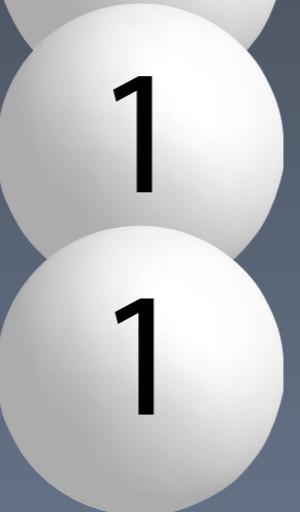
0



0



0



2



2



5



4



2



1



1



1



1



1



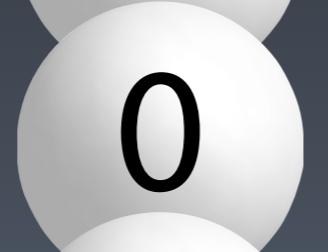
1



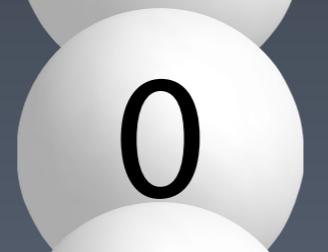
0



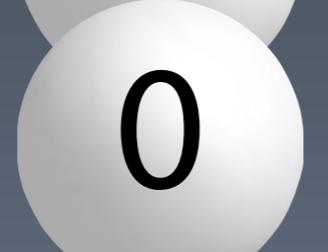
0



0



0



2



2



3



Counting ?

- 數（三聲）數（四聲）的概念
- 一個範圍太大，對應的範圍卻很小
- 重新對應簡化動作
- 再看一次？

程式碼

```
int numbers = 50;
int maxScore = 100;
int studentScore[50];
for (int i = 0; i < numbers; i++) {
    studentScore[i] = rand()%maxScore+1;
}
int score[101];
for (int i = 0; i < maxScore; i++) {
    score[i] = 0;
}

for (int i = 0; i < numbers; i++) {
    score[studentScore[i]] = score[studentScore[i]] + 1;
}
sort(score, maxScore+1);
```

程式碼

較大的數量

```
int numbers = 50;
int maxScore = 100;
int studentScore[50];
for (int i = 0; i < numbers; i++) {
    studentScore[i] = rand()%maxScore+1;
}
int score[101];
for (int i = 0; i < maxScore; i++) {
    score[i] = 0;
}

for (int i = 0; i < numbers; i++) {
    score[studentScore[i]] = score[studentScore[i]] + 1;
}
sort(score, maxScore+1);
```

程式碼

較大的數量

較小的數量

```
int numbers = 50;
int maxScore = 100;
int studentScore[50];
for (int i = 0; i < numbers; i++) {
    studentScore[i] = rand()%maxScore+1;
}
int score[101];
for (int i = 0; i < maxScore; i++) {
    score[i] = 0;
}

for (int i = 0; i < numbers; i++) {
    score[studentScore[i]] = score[studentScore[i]] + 1;
}
sort(score, maxScore+1);
```

程式碼

較大的數量

較小的數量

Counting

```
int numbers = 50;
int maxScore = 100;
int studentScore[50];
for (int i = 0; i < numbers; i++) {
    studentScore[i] = rand()%maxScore+1;
}
int score[101];
for (int i = 0; i < maxScore; i++) {
    score[i] = 0;
}
for (int i = 0; i < numbers; i++) {
    score[studentScore[i]] = score[studentScore[i]] + 1;
}
sort(score, maxScore+1);
```

程式碼

```
int numbers = 50;
int maxScore = 100;
int studentScore[50];
for (int i = 0; i < numbers; i++) {
    studentScore[i] = rand()%maxScore+1;
}
int score[101];
for (int i = 0; i < maxScore; i++) {
    score[i] = 0;
}
for (int i = 0; i < numbers; i++) {
    score[studentScore[i]] = score[studentScore[i]] + 1;
}
sort(score, maxScore+1);
```

較大的數量

較小的數量

Counting

Sort

The diagram illustrates the iterative steps of a binary search algorithm on a sorted array. It consists of five horizontal rows, each representing an iteration. The first four rows show the search space being narrowed down, while the fifth row shows the target value found.

- Row 1:** The array has 12 slots. The first 7 slots are white, and the remaining 5 slots are black, representing the search space.
- Row 2:** A curved arrow points from the center of the black region in Row 1 to the center of the black region in Row 2. The first 6 slots are white, and the next 3 slots are black.
- Row 3:** A curved arrow points from the center of the black region in Row 2 to the center of the black region in Row 3. The first 9 slots are white, and the next 3 slots are black.
- Row 4:** A curved arrow points from the center of the black region in Row 3 to the center of the black region in Row 4. The first 11 slots are white, and the last slot is black.
- Row 5:** The word "Binary Search" is written in large, bold, black font. A curved arrow points from the center of the black region in Row 4 to the center of the single black slot in Row 5, indicating the target value has been found.

Binary Search

二分搜的理由？

- 今天想找 A 班上的 3 號同學
- 全班正在玩大風吹，所以他不在第三個位置上
- 我們怎麼做才能找到他？

二分搜的理由？

- 今天想找 A 班上的 3 號同學
- 全班正在玩大風吹，所以他不在第三個位置上
- 我們怎麼做才能找到他？

一個一個問？

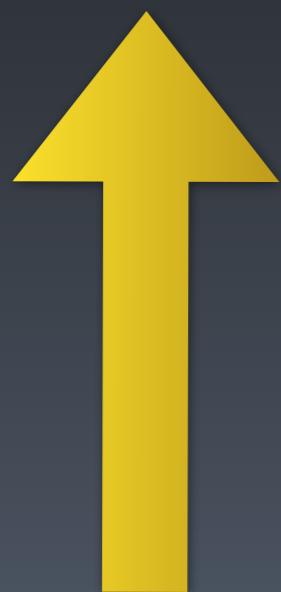
1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

1

5

4

2

3



你是 3 號嗎？

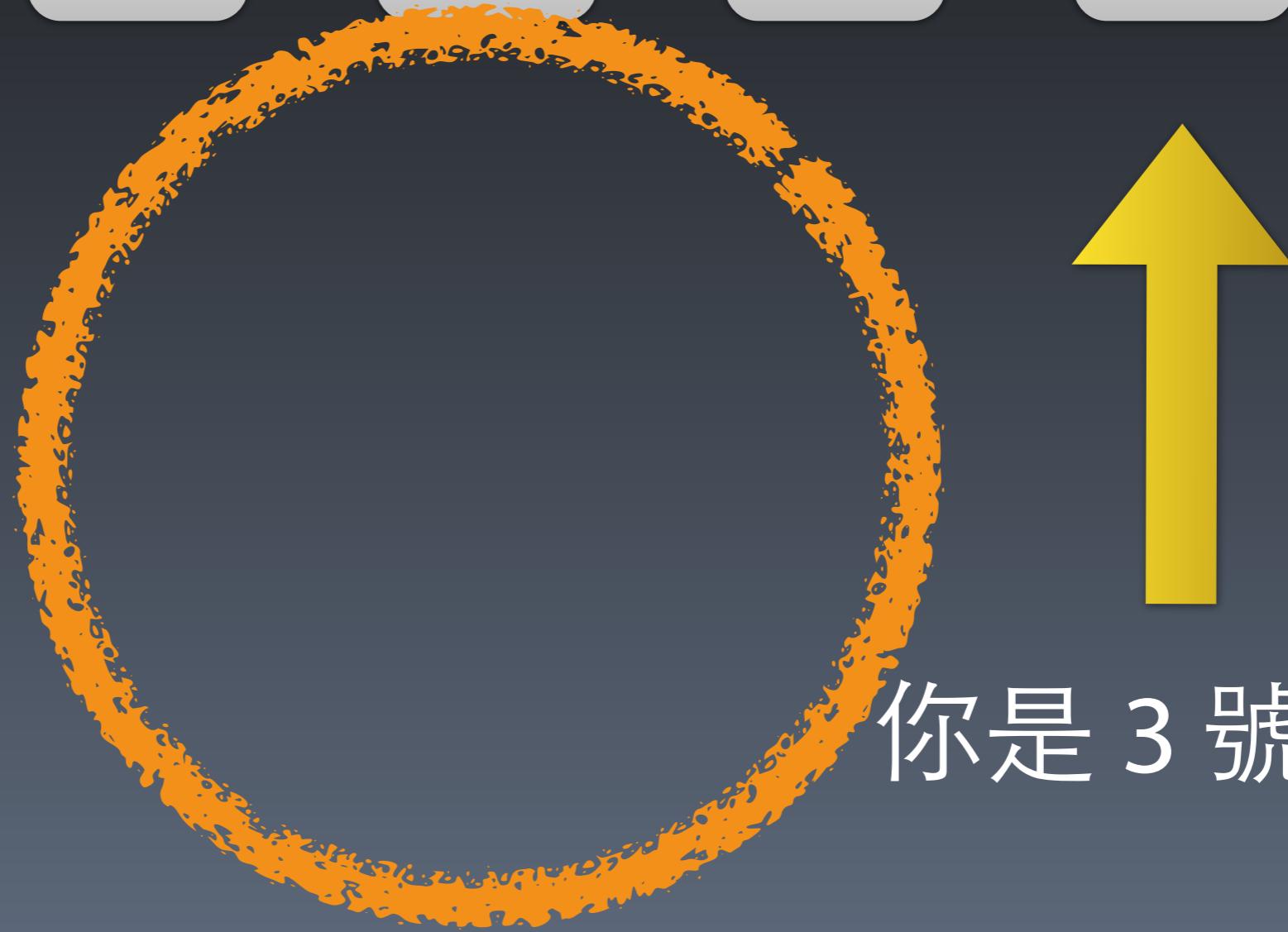
1

5

4

2

3



你是 3 號嗎？

老師表示：
給我排好！！！

1

1

2

3

4

1

1

2

3

4

3 號呢？！

1

1

3

4

2

3 號呢？！

1

1

3

4

3 號呢？！

二分搜

53

42

21

13

9

Wanted

13

53

42

13

9

21

Wanted

13

53

42

21

13

9

Wanted

13

53

42

21

13

9

Wanted

13

53

42

21

13

9

Wanted

13

53

42

21

9

13

Wanted

13

53

42

21

9

12

Wanted

13

53

42

21

12

9

Wanted

13

53

42

12

9

21

Wanted

13

53

42

21

12

9

Wanted

13

53

42

21

12

9

Wanted

13

53

42

21

12

9

Wanted

13

53

42

21

9

12

Wanted

13

53

42

21

12

9

Wanted

13

53

42

21

X

12

9

Wanted

13

53

42

21



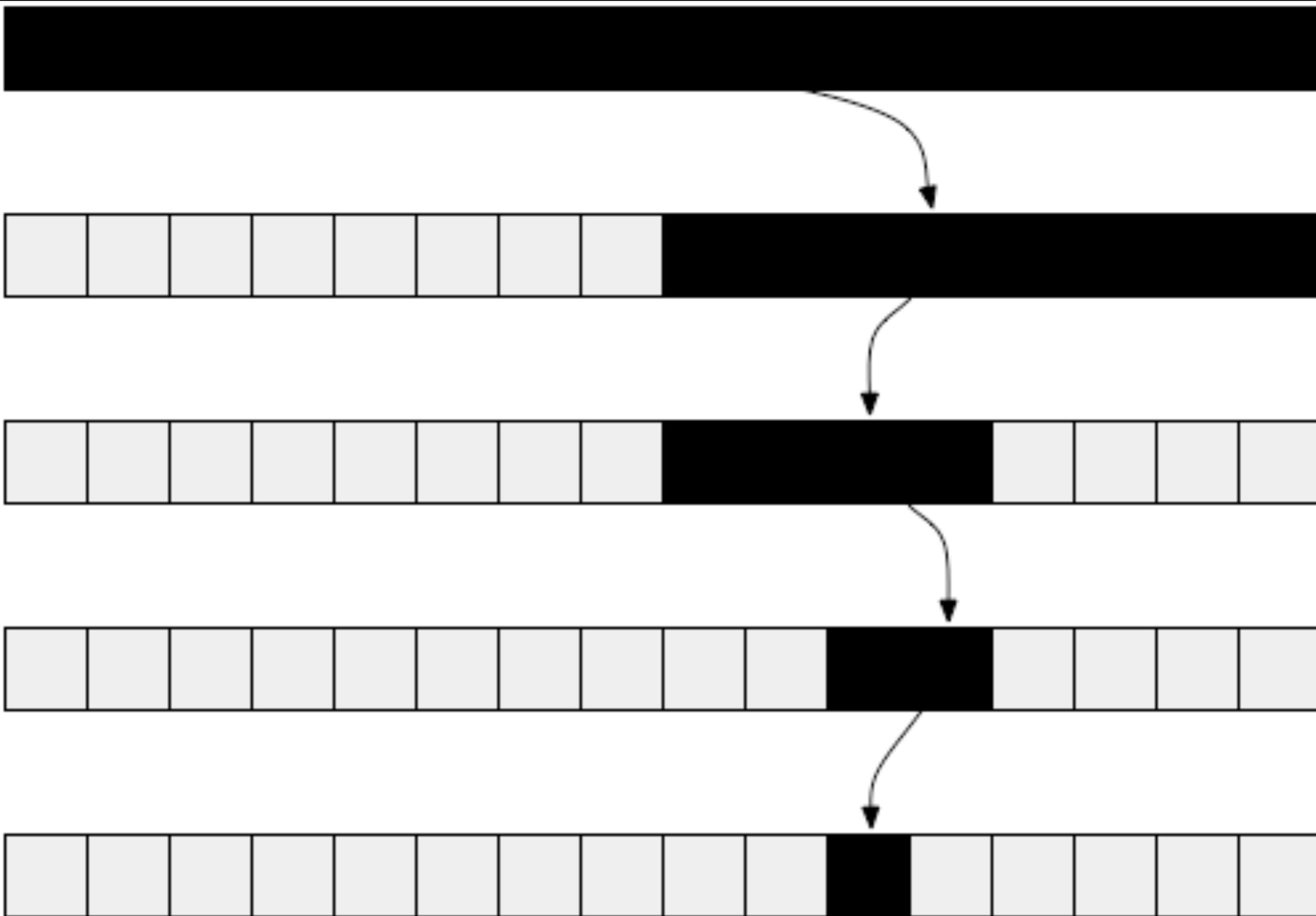
12

9

找不到東西 Q A Q

Wanted

13



二分搜

- *****超級無敵重要*****
- 忘記先排序，你就找不到了 Q A Q
- 每次都找剩下部分的中點
- 只看符合的那一半
- 到最後會收斂到一點 => 找到了 OWO !
- 到最後沒看到任何點 => 裡頭沒有 O T Z

程式碼

```
int Lower = 0;
int Upper = 100;
while (Lower <= Upper) {
    int Mid = (Lower+Upper)/2;
    if (data[Mid] == wantedData) {
        printf("Found!!!\n");
        break;
    } else if (data[Mid] > wantedData) {
        Upper = Mid - 1;
    } else if (data[Mid] < wantedData) {
        Lower = Mid + 1;
    }
}
```

作業

- Deadline
 - 2014/05/04 23:59
- 129 遲減的梵蒂岡聖光
- 130 全民大會考
- 131 時間魔術師
- 132 你在，不在