

NCKU Programming Contest Training Course

2017/03/22

Po-Hsing Wu

s1994928@yahoo.com.tw

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan



Outline

- **Binary Search**
- **Learn more**
 - **Operator overloading in struct**



Binary Search



Linear Search

- Example

– 從下方陣列中，找出小於 34 的最大值

| | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 6 | 25 | 14 | 13 | 84 | 51 | 43 | 53 | 64 | 97 | 33 | 96 | 95 | 93 | 72 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



Time Complexity

- Linear search : $O(n)$



Binary Search

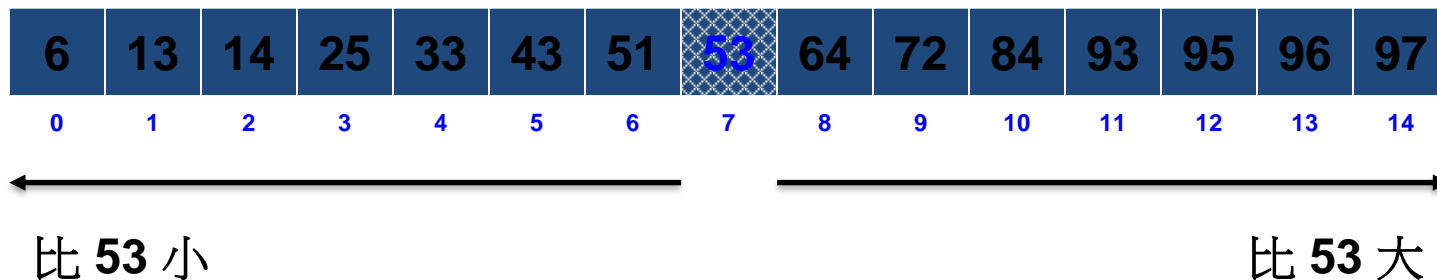
- 假設有個已經排序過後的序列

| | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 6 | 13 | 14 | 25 | 33 | 43 | 51 | 53 | 64 | 72 | 84 | 93 | 95 | 96 | 97 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |



Binary Search

- 假設有個已經排序過後的序列



Binary Search

- Example

– 從排序好的陣列中，找出小於 34 的最大值

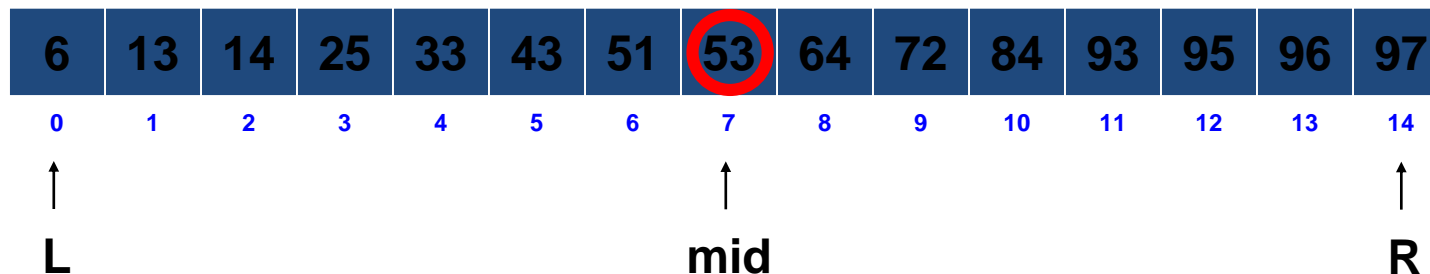
| | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 6 | 13 | 14 | 25 | 33 | 43 | 51 | 53 | 64 | 72 | 84 | 93 | 95 | 96 | 97 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| ↑ | | | | | | | | | | | | | | ↑ |
| L | | | | | | | | | | | | | | R |



Binary Search

- Example

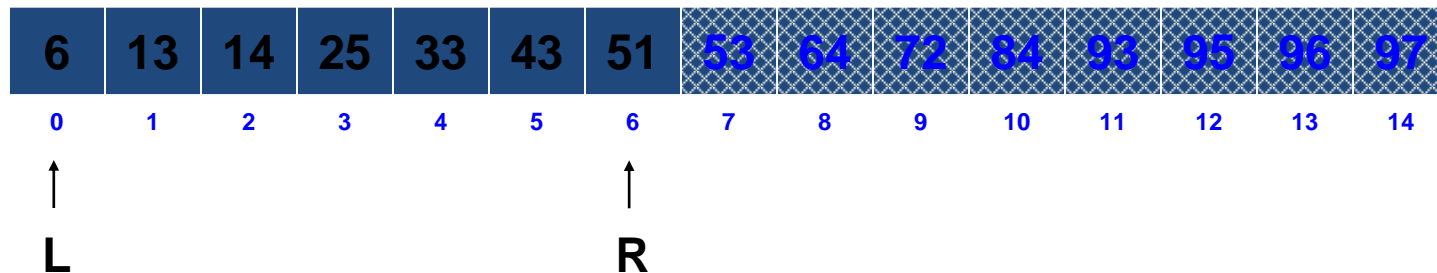
– Index of mid = $(L + R) / 2 = (0 + 14) / 2 = 7$



Binary Search

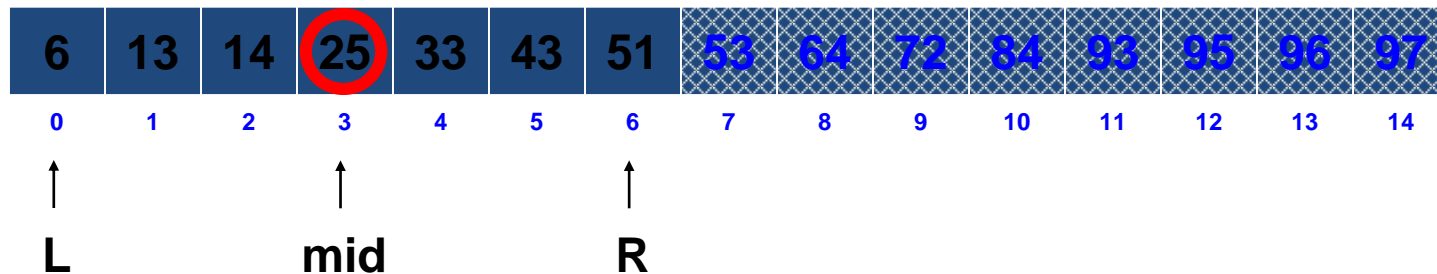
- Example

– 判斷 mid 有沒有小於 34 → 沒有，找左半部



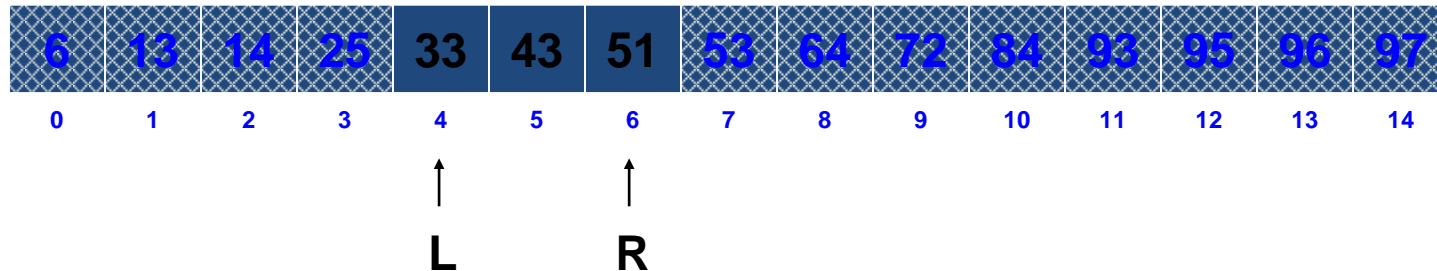
Binary Search

- Example
 - Index of mid = $(L + R) / 2 = (0 + 6) / 2 = 3$



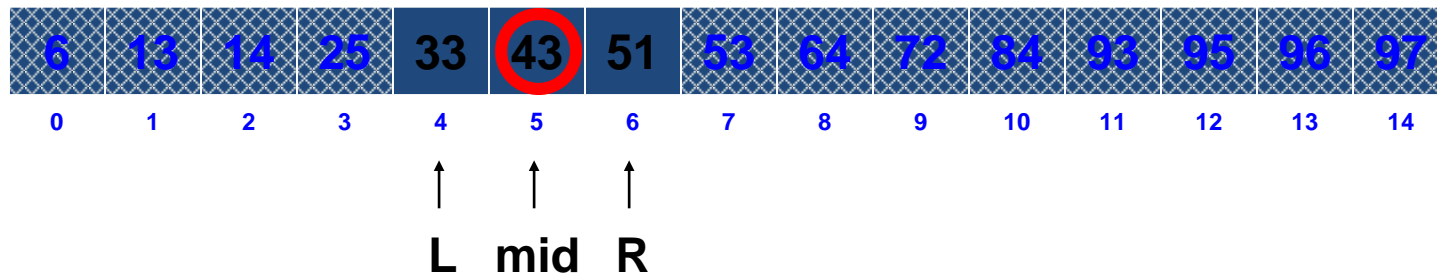
Binary Search

- Example
 - 判斷 mid 有沒有小於 34 → 有，找右半部



Binary Search

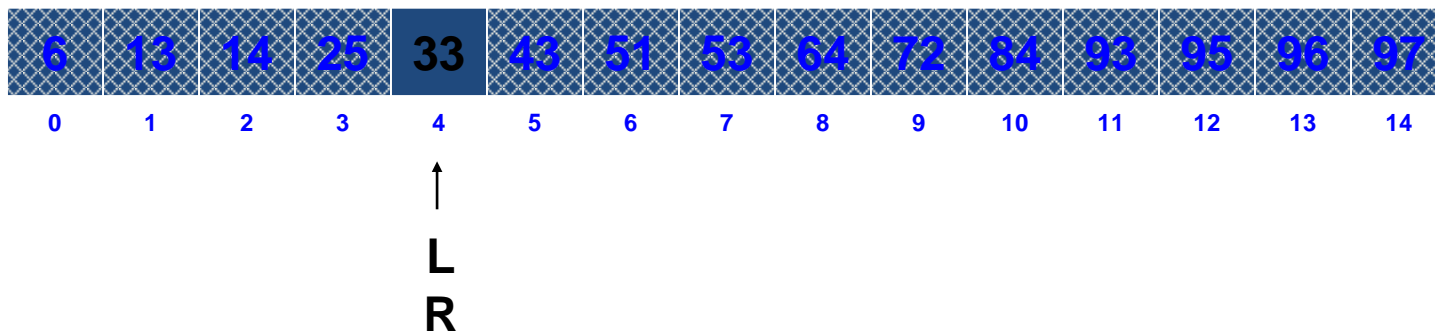
- Example
 - Index of mid = $(L + R) / 2 = (4 + 6) / 2 = 5$



Binary Search

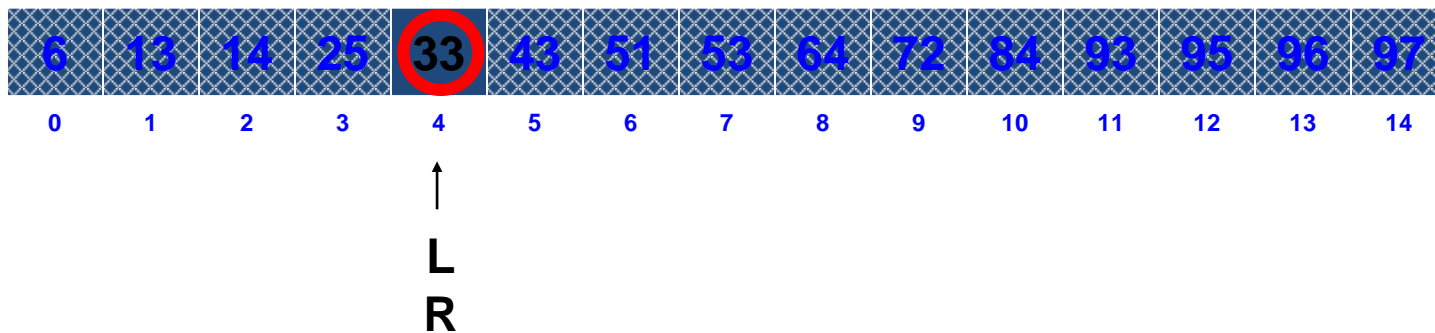
- Example

– 判斷 mid 有沒有小於 34 → 沒有，找左半部



Binary Search

- Example
 - 小於 34 的最大值 = 33



Time Complexity

- Linear search : $O(n)$
- Binary search : $O(\lg n)$



Binary Search

```
#include <stdio>

int binary_search(int *numbers, int n, int val) {
    int left = 0, right = n - 1;
    while(left < right) {
        int middle = (left + right) / 2;
        if (numbers[middle] < val) {
            left = middle + 1;
        } else {
            right = middle - 1;
        }
    }

    return right;
}

int main() {
    int array[] = {6, 13, 14, 25, 33, 43, 51, 53, 64, 72, 84, 93, 95, 96, 97};
    printf("%d\n", binary_search(array, 15, 34));    // 4

    return 0;
}
```



UVa 10341

Problem Description

Solve

$$p * e^{-x} + q * \sin(x) + r * \cos(x) + s * \tan(x) + t * x^2 + u = 0$$

Where

$$0 \leq p, r \leq 20 \text{ and } -20 \leq q, s, t \leq 0$$

$$0 \leq x \leq 1$$



lower/upper_bound

- Binary Search 不用自己寫，STL 替你完成一切

```
#include <cstdio>
#include <algorithm>

using namespace std;

int main() {
    int array[] = {6, 13, 14, 25, 33, 43, 51, 53, 64, 72, 84, 93, 95, 96, 97};
    printf("%d\n", lower_bound(array, array + 15, 34) - array - 1);

    return 0;
}
```



Learn more: Operator overloading in struct



struct

Q. 如何直接利用 < 來比較 struct ?

A. 運算子重載

```
struct _DATA {  
    int data1, data2;  
    char data3;  
  
    bool operator<(const struct _DATA &rhs) const {  
        return data1 < rhs.data1;  
    }  
};
```



struct

Q. 如何用 lower/upper_bound 搜尋 struct ?

A. 運算子重載

```
struct _DATA {  
    int data1, data2;  
    char data3;  
  
    bool operator<(const int val) const {  
        return data1 < val;  
    }  
};
```

