Binary Search

Jyh-Shing Roger Jang (張智星)

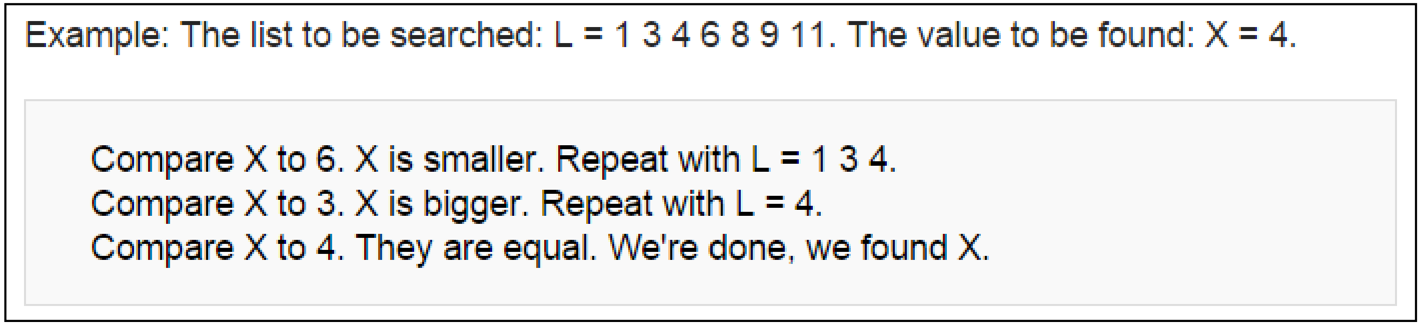
CSIE Dept, National Taiwan University

Binary Search Algorithm

- Definition
  - Find the position of a specified input key value within an array sorted by key value

Keyword: Sorted!

- Daily life example
  - Find a specific page in a book (翻書找頁碼)
  - Search for a word in a dictionary (從字典中找單字)
- Numerical example

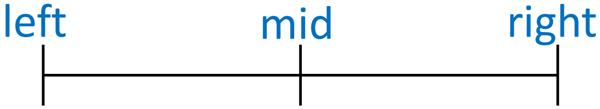


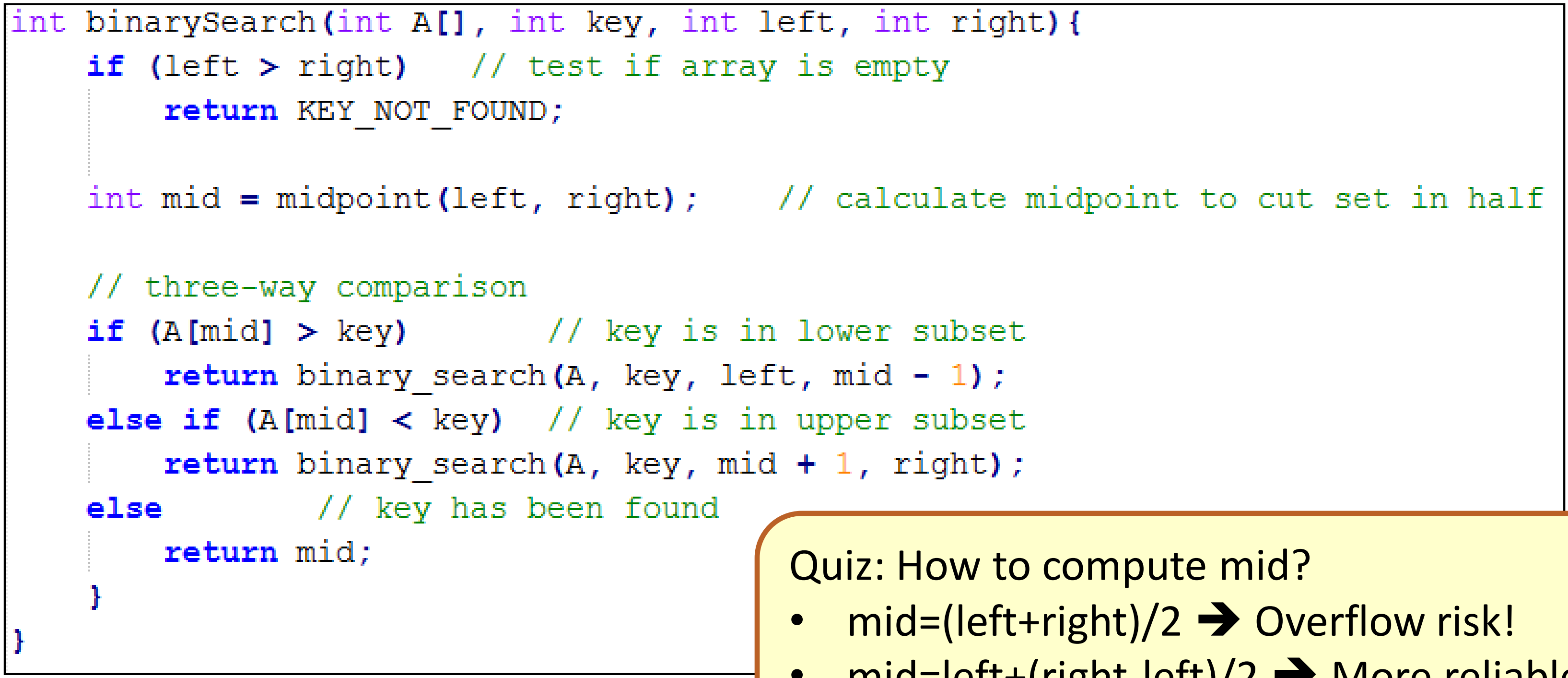
Example: The list to be searched: L = 1 3 4 6 8 9 11. The value to be found: X = 4.

```
Compare X to 6. X is smaller. Repeat with L = 1 3 4.
Compare X to 3. X is bigger. Repeat with L = 4.
Compare X to 4. They are equal. We're done, we found X.
```

Recursive Function for Binary Search

- Recursive function





```
int binarySearch(int A[], int key, int left, int right){
    if (left > right)    // test if array is empty
        return KEY_NOT_FOUND;

    int mid = midpoint(left, right);    // calculate midpoint to cut set in half

    // three-way comparison
    if (A[mid] > key)       // key is in lower subset
        return binary_search(A, key, left, mid - 1);
    else if (A[mid] < key)  // key is in upper subset
        return binary_search(A, key, mid + 1, right);
    else          // key has been found
        return mid;
    }
}
```

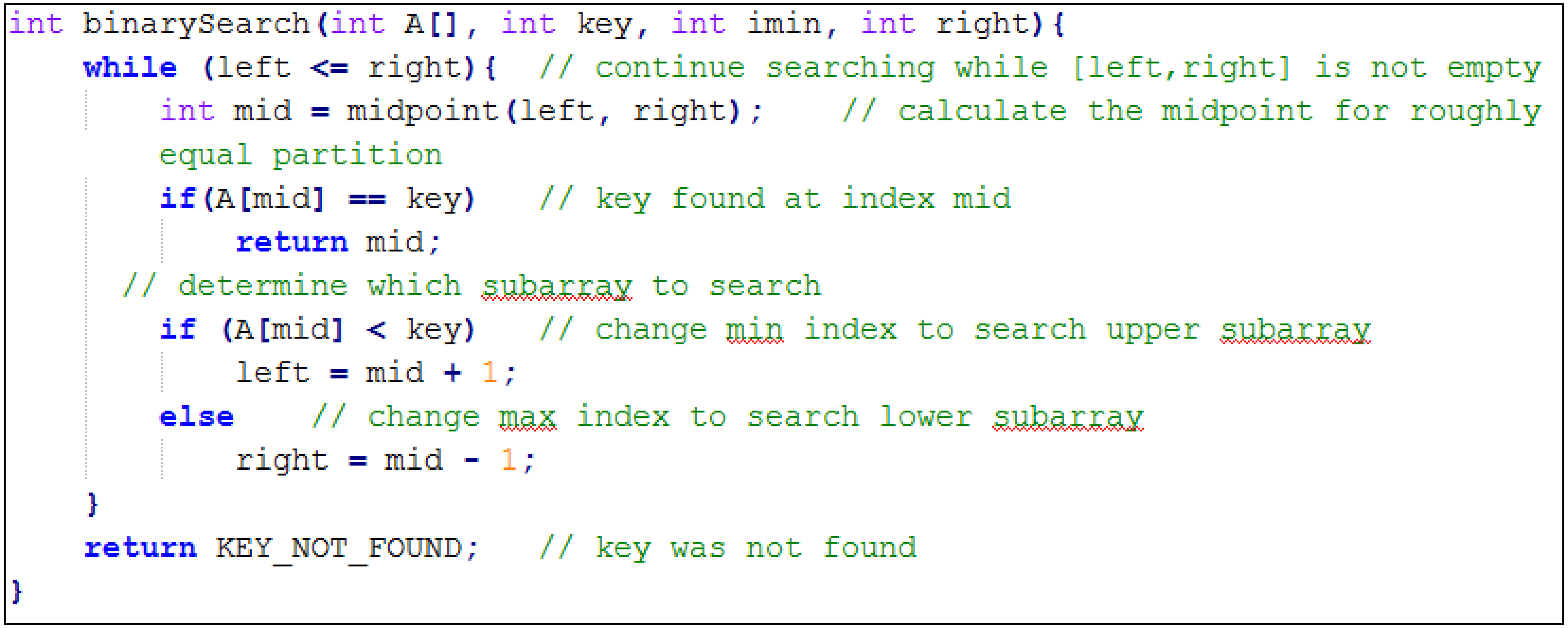
Quiz: How to compute mid?

- mid=(left+right)/2 ➔ Overflow risk!
- mid=left+(right-left)/2 ➔ More reliable!
- mid=left/2+right/2 ➔ Slower???

- Example usage
  - index = binarySearch(vec, key, 0, vec.size()-1)

Iterative Function for Binary Search

- Iterative function



```
int binarySearch(int A[], int key, int imin, int right){
    while (left <= right){  // continue searching while [left,right] is not empty
        int mid = midpoint(left, right);    // calculate the midpoint for roughly
        equal partition
        if(A[mid] == key)   // key found at index mid
            return mid;
      // determine which subarray to search
        if (A[mid] < key)   // change min index to search upper subarray
            left = mid + 1;
        else     // change max index to search lower subarray
            right = mid - 1;
    }
    return KEY_NOT_FOUND;   // key was not found
}
```

Quiz: A better way to compute mid
➔ By interpolation

Summary

- Comparisons

  Quiz!

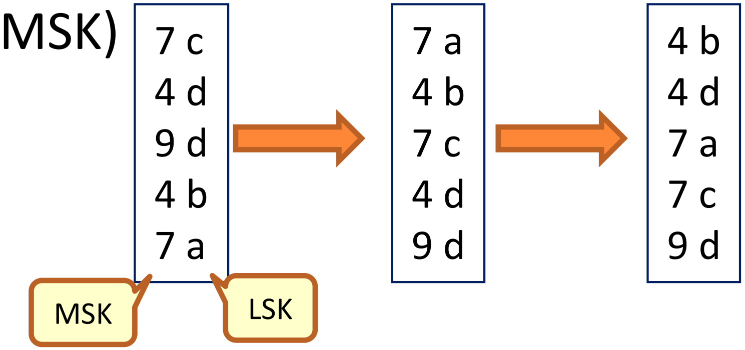
  - Linear search
    - Complexity O(n)
    - For any unsorted arrays ➔ Fast when appending new elements Considerable speedup if frequently searched items are placed at the beginning
  - Binary search
    - Complexity O(log(n))
    - For sorted arrays ➔ Slow when inserting new elements
  - Hash search
    - Complexity O(1)
    - For arrays pre-processed by hash functions

Extensions

- Other similar problems
  - Interval finding (e.g., to speed up insertion sort)
    - Given a sorted vector, find the interval of a given value.
  - Non-zero element finding
    - Given a sign-sorted vector, find the no. of positive elements.
- Binary search using multiple keys

Quiz!

  - Preprocessing stage: Stable-sort with multiple keys, starting from the least-significant key (LSK)
  - Search stage: Binary search starting from the most-significant key (MSK)

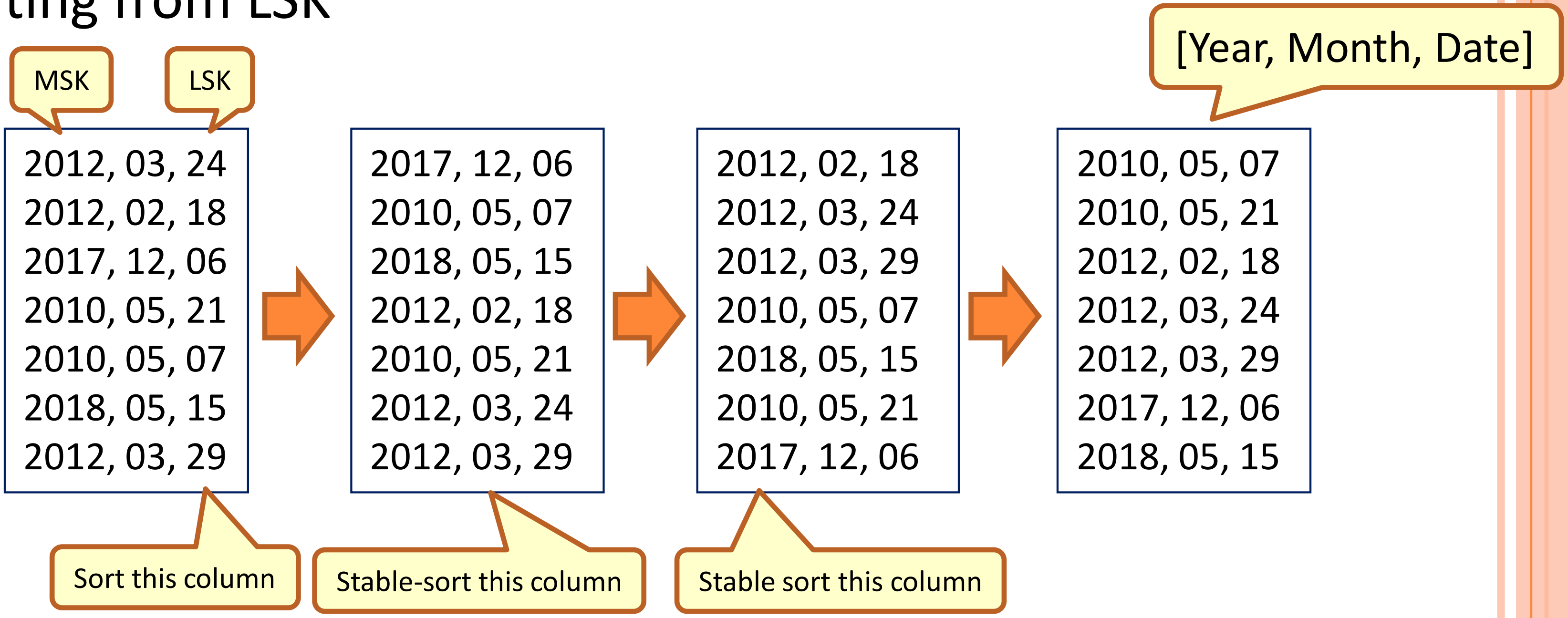


7 c	7 a	4 b
4 d	4 b	4 d
9 d	7 c	7 a
4 b	4 d	7 c
7 a	9 d	9 d

Example of Binary Search with Multiple Keys

Quiz!

- Preprocessing stage: Stable-sort data with multiple keys, starting from LSK



MSK

LSK

[Year, Month, Date]

2012, 03, 24
2012, 02, 18
2017, 12, 06
2010, 05, 21
2010, 05, 07
2018, 05, 15
2012, 03, 29

2017, 12, 06
2010, 05, 07
2018, 05, 15
2012, 02, 18
2010, 05, 21
2012, 03, 24
2012, 03, 29

2012, 02, 18
2012, 03, 24
2012, 03, 29
2010, 05, 07
2018, 05, 15
2010, 05, 21
2017, 12, 06

2010, 05, 07
2010, 05, 21
2012, 02, 18
2012, 03, 24
2012, 03, 29
2017, 12, 06
2018, 05, 15

Sort this column

Stable-sort this column

Stable sort this column

- Search stage: Binary search starting from MSK
  - Let’s search for (2012, 03, 29)