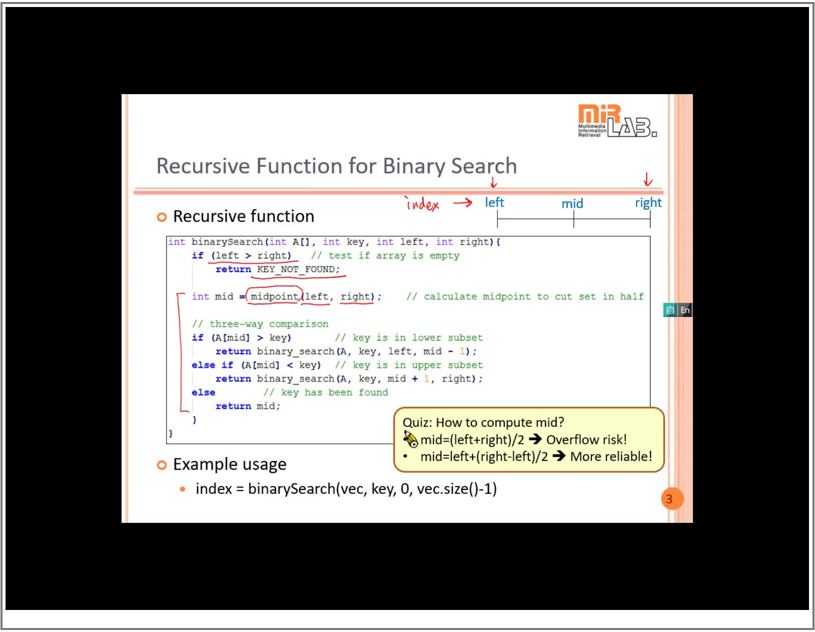


Recursive Function for Binary Search

- 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;
    }
}
```

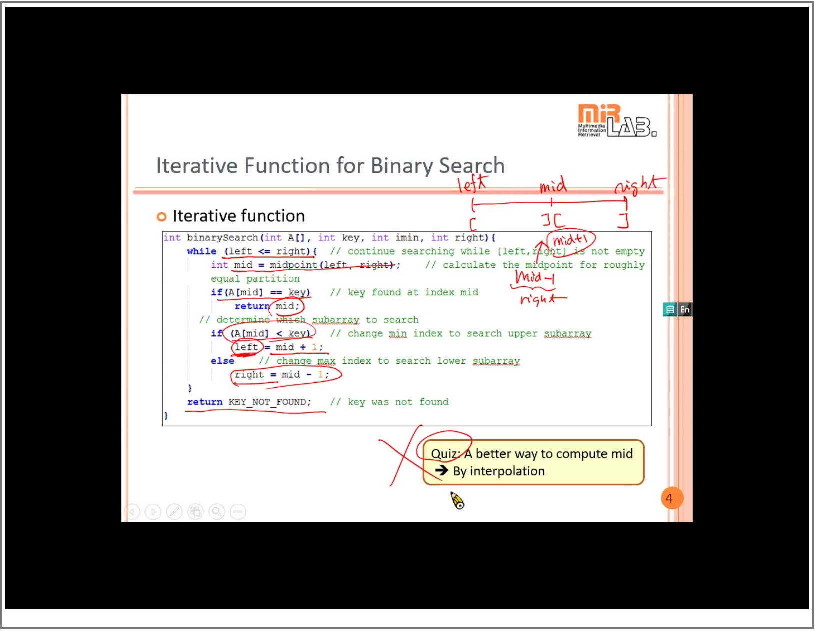
Quiz: How to compute mid?
- mid=(left+right)/2 ➔ Overflow risk!
- mid=left+(right-left)/2 ➔ More reliable!

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

3

(6:49) by Oliver Chang

How to compute mid?



Iterative Function for Binary Search

- Iterative function

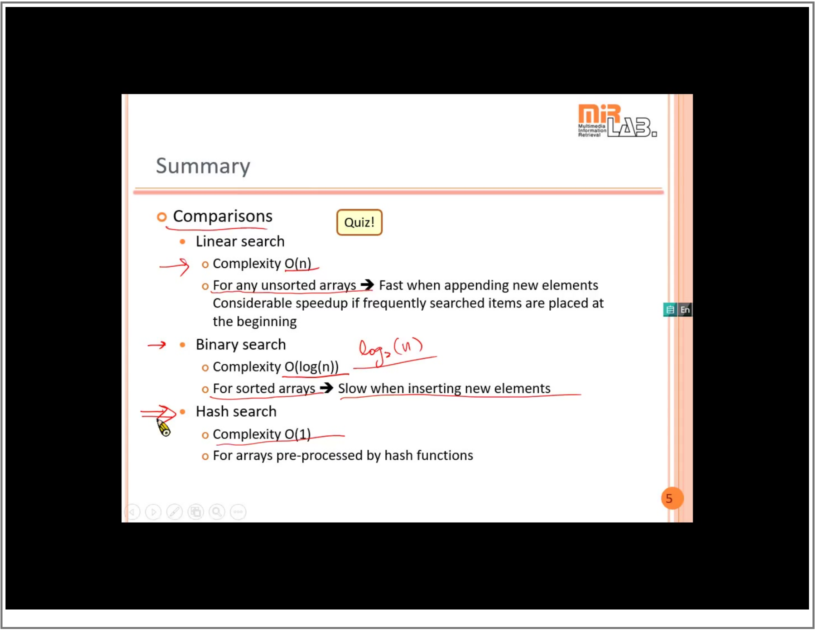
```
int binarySearch(int A[], int key, int imin, int right){
    while (left <= right){  // continue searching while [left,right] is not empty
        int mid = midpoint(left, right);    // calculate the midpoint for roughly
        equal partition
        if(A[mid] == key)   // key found at index mid
            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

4

(16:15) by Oliver Chang

A better way to compute mid
-> by Interpolation (內插)



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

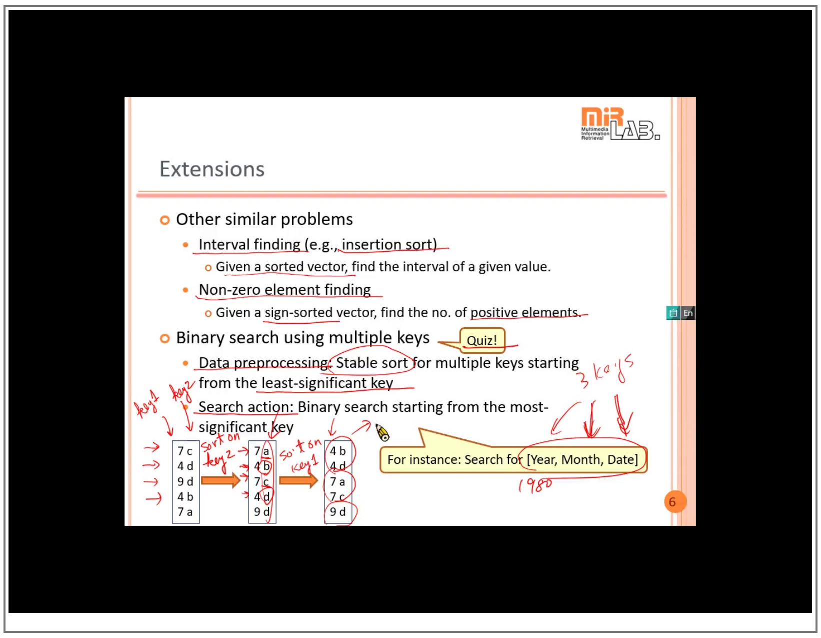
5

(21:22) by Oliver Chang

Linear
Binary
Hash

(29:45) by Oliver Chang

Stable Sort



Extensions

- Other similar problems
  - Interval finding (e.g., 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!
  - Data preprocessing: Stable sort for multiple keys starting from the least-significant key
  - Search action: Binary search starting from the most-significant key

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

For instance: Search for [Year, Month, Date]