Search & Select Algorithms

Data Structures and Algorithms

Sep. 2nd, 2014

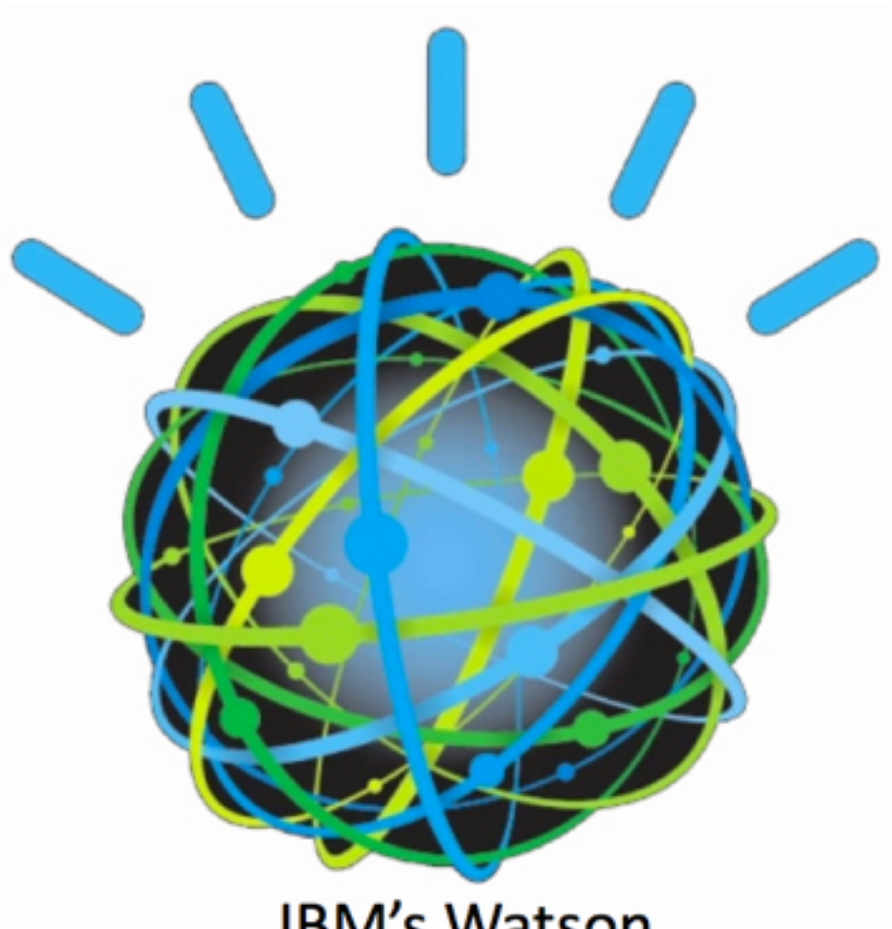
Jinho D. Choi

EMORY
UNIVERSITY

Administrative

- Class enrollment
  - Last day for add and drop → Sep. 3rd.
  - Audit the course.
  - Piazza.
- Exercises
  - Java and Eclipse installation.
  - https://github.com/jdchoi77/emory-courses/wiki
  - Please keep up with the exercises. Each exercise assumes you've complicated the previous ones.
- The class will be at E308A from Thursday (Sep. 4th).

Watson For Researchers



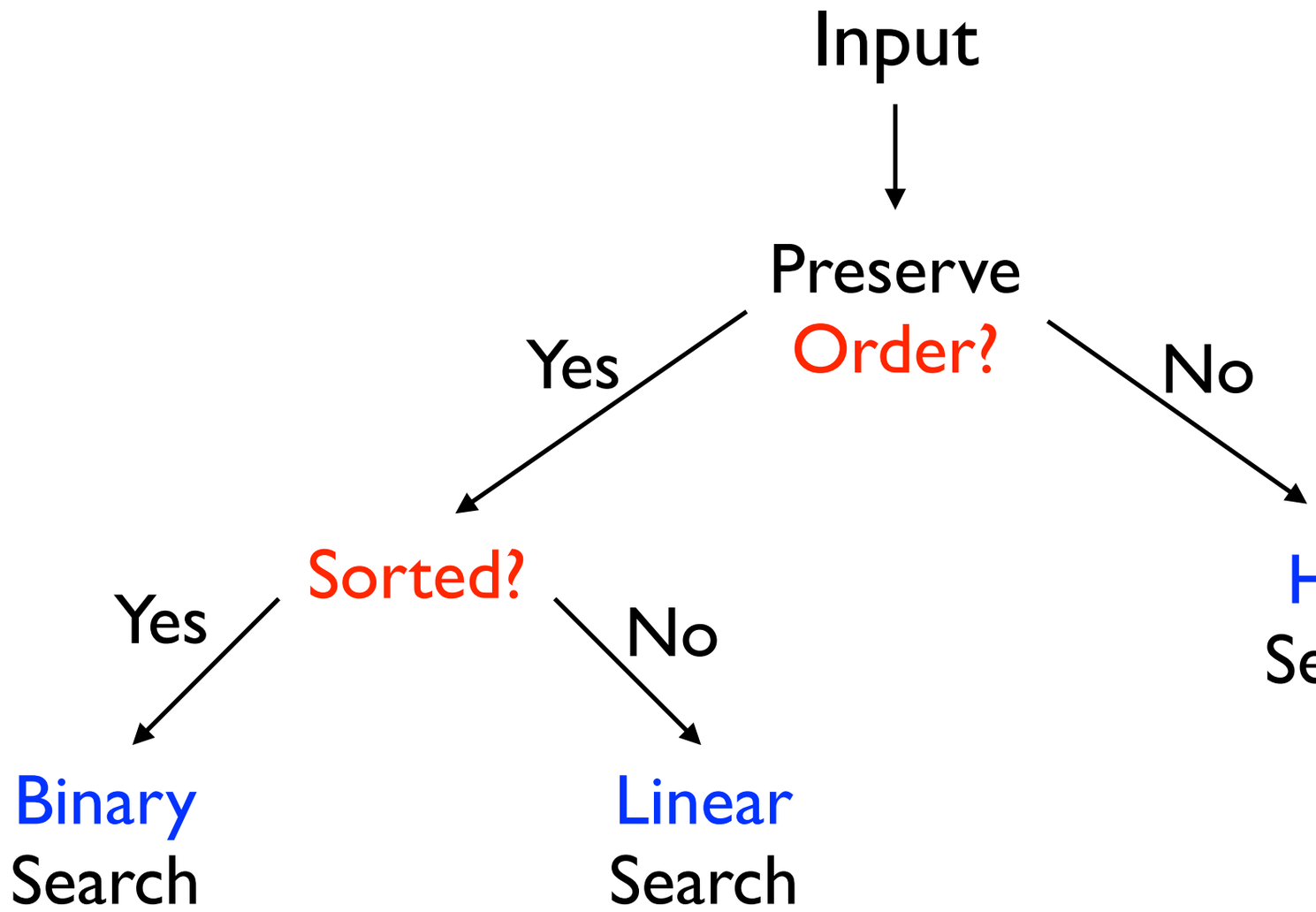
IBM's Watson

https://www.youtube.com/watch?v=yJptrICVDHI

http://venturebeat.com/2014/07/18/inside-ibms-billion-dollar-bet-on-watson/

http://venturebeat.com/2014/08/27/ibm-finally-opens-up-its-watson-supercomputer-to-researchers/

Search Algorithm



Indexing?

Caching?

Threading?

Search Interface

Class? Generics? Comparable?

```
public interface ISearch<T extends Comparable<T>>
{
   /**
    * @param list a list containing zero to many keys.
    * @param key a key to be searched.
    * @return the index of the key in the list if exists;
    *          otherwise, a negative integer.
    */
   int search(List<T> list, T key);
}
```

This is called?

Javadoc

EMORY UNIVERSITY

Linear Search

```
public int search(List<T> list, T key)
{
  int i, size = list.size();

  for (i=0; i<size; i++)
  {
    if (key.equals(list.get(i)))
      return i;
  }

  return -1;
}
```

Complexity?

Possible to be faster than $O(n)$?

EMORY UNIVERSITY

Binary Search

```
protected int search(List<T> list, T key, int beginIndex, int endIndex)
{
  if (beginIndex > endIndex)
    return -1;
  else if (beginIndex == endIndex)
    return key.equals(list.get(beginIndex)) ? beginIndex : -1;

  int middleIndex = beginIndex + (endIndex - beginIndex) / 2;
  int diff = key.compareTo(list.get(middleIndex));

  if (diff > 0)
    return search(list, key, middleIndex+1, endIndex);
  else if (diff < 0)
    return search(list, key, beginIndex, middleIndex-1);
  else
    return middleIndex;
}
```

Necessary?

Complexity?

Duplicated keys?

EMORY UNIVERSITY

Quiz

- See BinarySearch.java.
  - https://github.com/jdchoi77/emory-courses/blob/master/src/main/java/edu/emory/mathcs/cs323/search/BinarySearch.java
- Imagine that the following condition didn't exist (line 38~).

```
if (beginIndex > endIndex)
  return -1;
```

  - What kind of exception would this program throw if an empty list were inserted to the search method?
  - Give a list (not empty, sorted in ascending order) that would make this program throw a different kind of exception. Explain why this exception would be thrown.
- Post your answer to Piazza before next Tuesday (Sep. 9th).

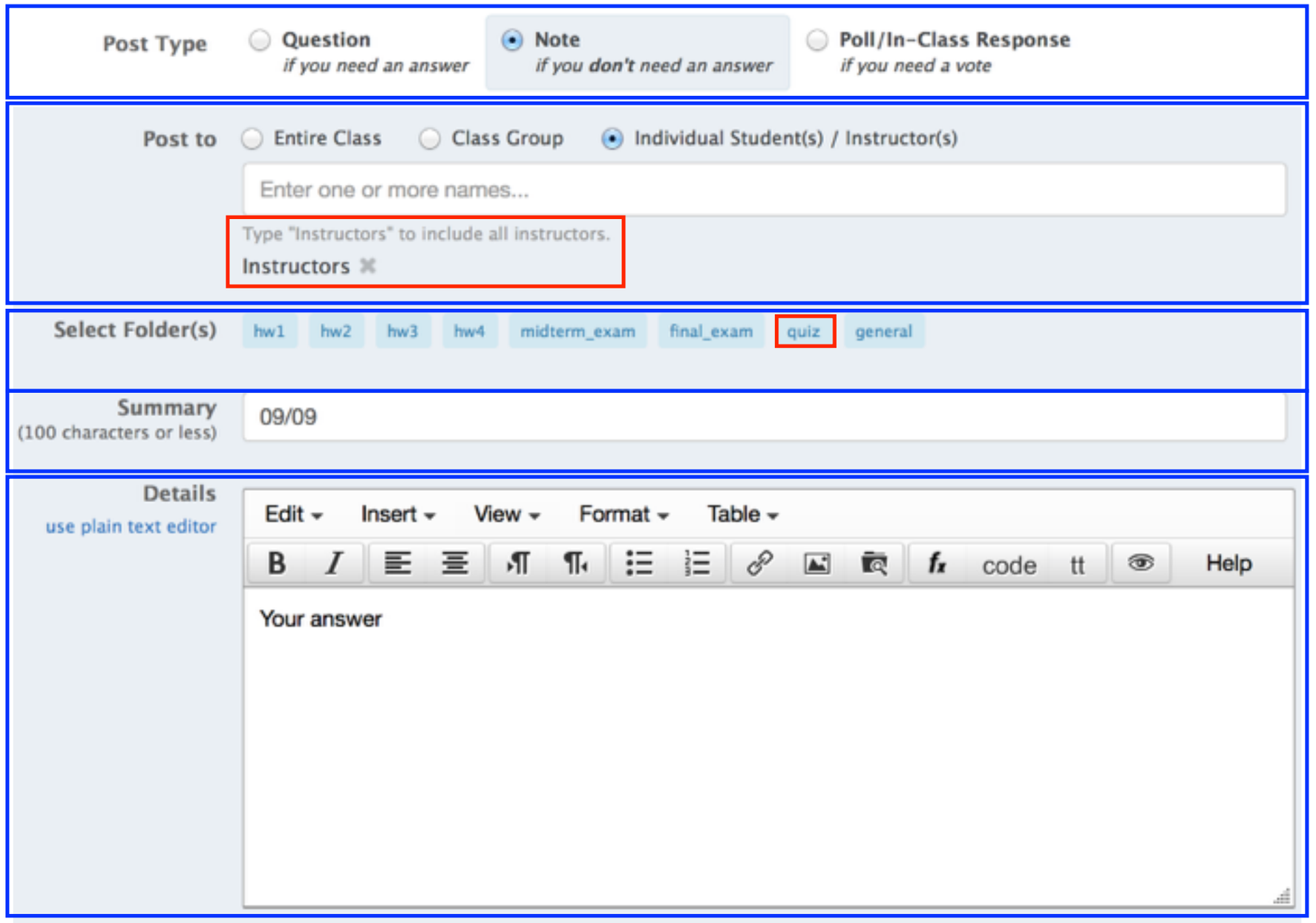
How To Submit

Question

Instructors

Quiz

Due date



Your Answer

Accuracy Test

- You must ensure the correctness of your program.
  - Unit test is one way of doing it.
  - The more test cases you have, the more robust your program becomes.

```
public void testAccuracy()
{
  ISearch<Integer> s = new BinarySearch<>();
  List<Integer> list = DSUtils.toIntegerList(5, 2, 3, 2, 1, 1, 4, 2);
  Collections.sort(list);  // list = [1, 1, 2, 2, 2, 3, 4, 5]

  assertEquals(s.search(list, 1), 1);
  assertEquals(s.search(list, 2), 3);
  assertEquals(s.search(list, 3), 5);
  assertEquals(s.search(list, 5), 7);
  assertTrue(s.search(list, 0) < 0);
}
```

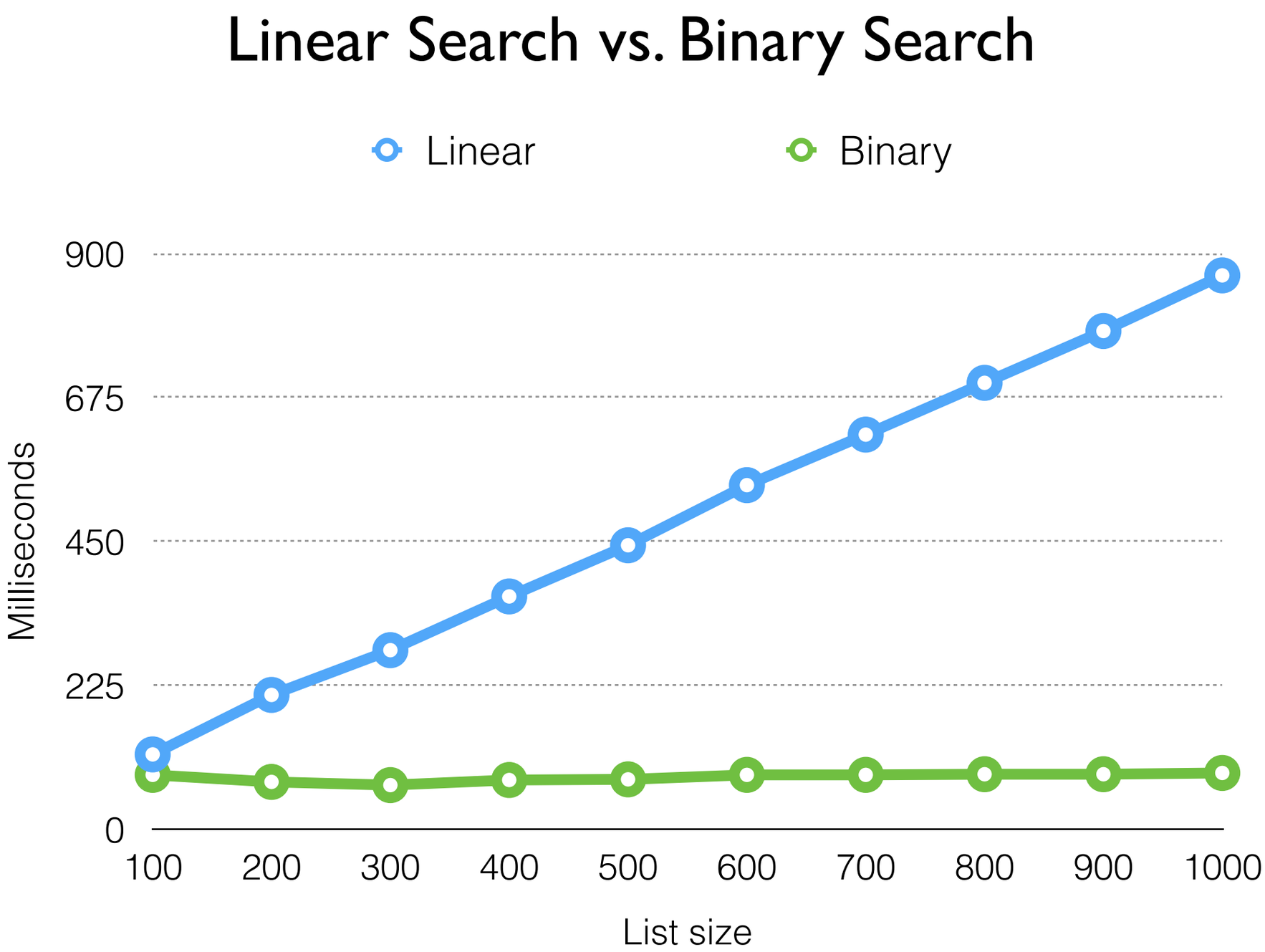
Is this algorithm stable?

Speed Test

- If you want to see if your program runs as fast as it is designed,
  - The best way is using a profiler (e.g., JProfiler).
  - The easiest way is measuring the time between operations.

```
public long getRuntime(int iterations)
{
  long st = System.currentTimeMillis();

  for (int i=0; i<iterations; i++)
  {
    // Operations ...
  }

  long et = System.currentTimeMillis();

  return et - st;
}
```

Speed Test


Linear Search vs. Binary Search
Linear
Binary
900
675
450
225
0
Milliseconds
100
200
300
400
500
600
700
800
900
1000
List size

Select Algorithm

- Find the maximum key in a list
  - $O(n)$.
- Find the *k*'th maximum key in a list
  - Complexity?
- Dumb select
  - Make a copy of the input list.
  - Find and remove the maximum key from the copied list.
  - Repeat this procedure *k* times, and return the last maximum key.

Select Abstract Class

vs. Interface?

```
public abstract class AbstractSelect<T extends Comparable<T>>
{
  /**
   * @param list a list of unsorted keys.
   * @param k the k'th maximum key to search.
   * @return the k'th maximum key in the list.
   * @throws IllegalArgumentException if the size of the list is smaller than k.
   */
  abstract public T max(List<T> list, int k);

  protected void throwIllegalArgumentException(List<T> list, int k)
  {
    if (list.size() < k)
      throw new IllegalArgumentException("The array size is smaller than k.");
  }
}
```

EMORY UNIVERSITY

Dumb Select

```
public T max(List<T> list, int k)
{
  throwIllegalArgumentException(list, k);
  List<T> copy = new ArrayList<>(list);
  T max = null;

  for (int i=0; i<k; i++)
  {
    max = Collections.max(copy);
    copy.remove(max);
  }

  return max;
}
```

$O(k \cdot n)$

How many comparisons?

Smart Select

- Smart select
  - Create a new empty list, *L*.
  - For each key in the input list,
    - Add the key to *L*.
    - If the size of *L* is greater than *k*, remove the minimum key in *L*.
  - Return the minimum key in *L*.

- Group discussions
  - Design an efficient algorithm for "Smart Search".
  - What is the worst-case complexity (how many comparisons)?

Smart Select

$$k = 3$$

Smart Select

$k = 3$

maxK = 

Smart Select

$k = 3$

maxK = | | | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

EMORY
UNIVERSITY

Smart Select

$k = 3$

maxK = | | | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 3 | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 3 | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 3 | 2 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 3 | 2 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 3 | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 4 | 3 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 4 | 3 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 4 | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 6 | 4 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 6 | 4 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 6 | 4 | 1

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 6 | 4 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 6 | 4 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 6 | |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

$k = 3$

maxK = | 7 | 6 | 5 |

List = | 7 | 3 | 2 | 4 | 6 | 1 | 5 |

Smart Select

```
public T max(List<T> list, int k)
{
  List<T> maxK = new ArrayList<>(k);
  for (T item : list) insert(maxK, item, k);
  return maxK.get(maxK.size()-1);
}
```

$O(n \cdot log\ k)$

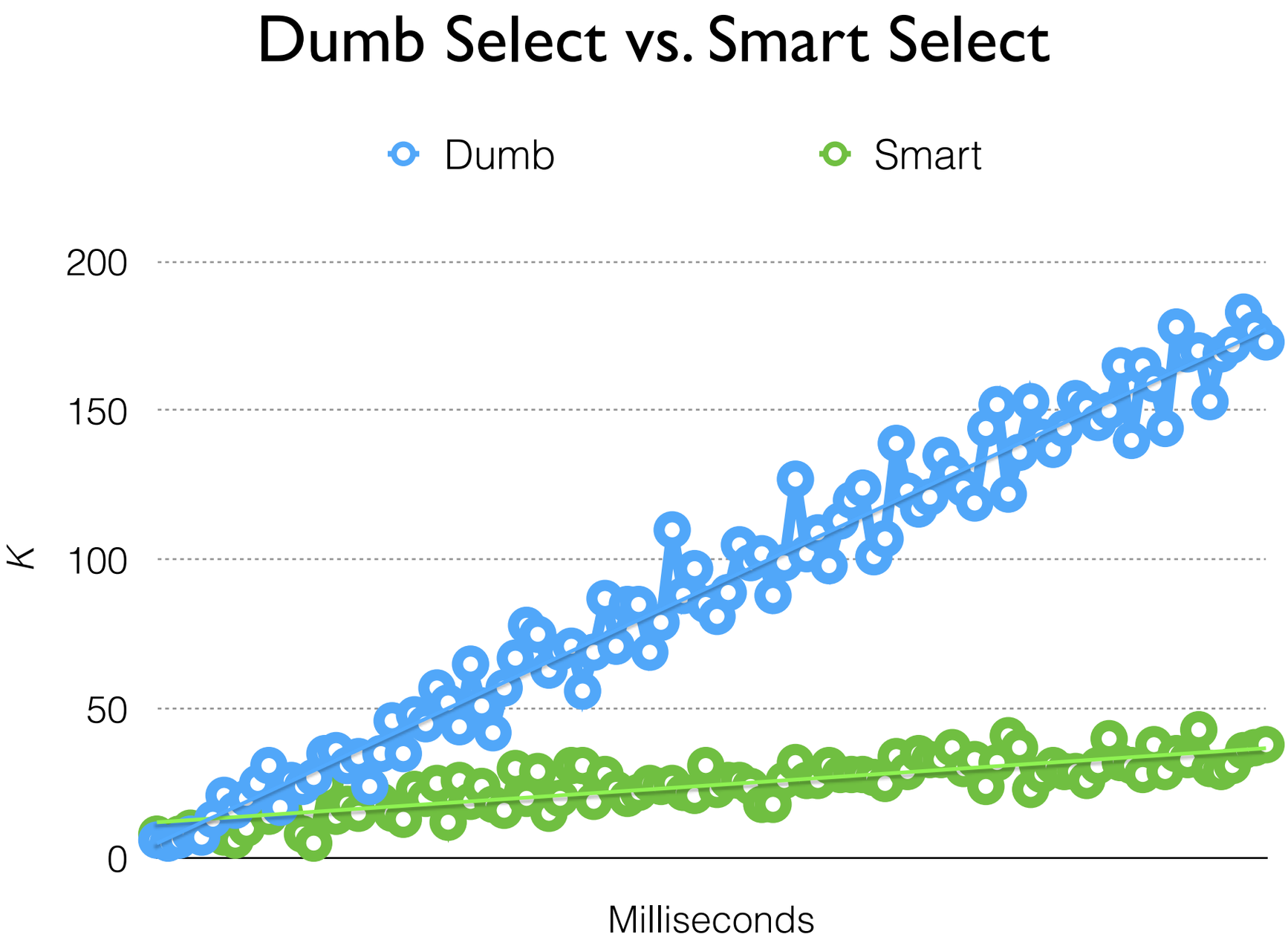
```
private void insert(List<T> maxK, T item, int k)
{
  int index = Collections.binarySearch(maxK, item, Collections.reverseOrder());
  if (index < 0) index = -(index + 1);

  if (index < k)
  {
    maxK.add(index, item);
    if (maxK.size() > k) maxK.remove(maxK.size()-1);
  }
}
```

How many comparisons?

Is this really faster?

EMORY UNIVERSITY

Smart Select



Before Next Class

- Exercises
  - Search and select.
  - https://github.com/jdchoi77/emory-courses/wiki
- Quiz
  - See the slides 7.
  - Submit your answer before the next class (Sep. 9th).
  - You are responsible for making sure if there are quizzes in each class.
- Reading
  - Chapter 2.4 - Priority Queues.