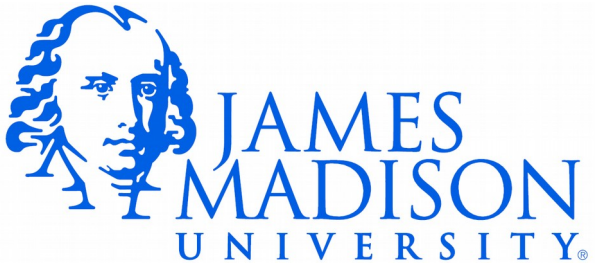
CS139 Binary Search



Puzzle Lab

- Let's look at the solution code...

Why Is It So SLOW?

- 263,533 words
- 13,712 8-letter words ending in 's'
- 13,712 * 263,533 = 3,613,564,496
- We need to make about 3 billion string comparisons.
  - Each string comparison may involve multiple character comparisons
- Is there a better way???

Binary Search

- http://www.cs.armstrong.edu/liang/animation/web/BinarySearch.html
- http://balance3e.com/Ch8/search.html

Binary Search Code

```java
/**
 * Search the provided array for the target value.  Return -1 if
 * target is not present.  The array must be sorted.
 */
public static int search(int[] numbers, int target) {

   int start = 0;
   int end = numbers.length - 1;
   int mid;

   while (start <= end) {
      mid = (start + end) / 2;

      if (target == numbers[mid]) {
         return mid;
      } else if (target < numbers[mid]) {
         end = mid - 1;
      } else {
         start = mid + 1;
      }
   }

   return -1;
}
```

Binary Search Exercise

```
public static int search(int[] numbers, int target) {

   int start = 0;
   int end = numbers.length - 1;
   int mid;

   while (start <= end) {
      mid = (start + end) / 2;

      System.out.println(start + " " + mid + " " + end);

      if (target == numbers[mid]) {
         return mid;
      } else if (target < numbers[mid]) {
         end = mid - 1;
      } else {
         start = mid + 1;
      }
   }

   return -1;
}
```

What will be printed when this code executes?

```
int[] values = { 0, 26, 55, 82, 91, 104, 112, 124 };

System.out.println(search(values, 91));
```

Binary Search Exercise

```
public static int search(int[] numbers, int target) {

   int start = 0;
   int end = numbers.length - 1;
   int mid;

   while (start <= end) {
      mid = (start + end) / 2;

      System.out.println(start + " " + mid + " " + end);

      if (target == numbers[mid]) {
         return mid;
      } else if (target < numbers[mid]) {
         end = mid - 1;
      } else {
         start = mid + 1;
      }
   }

   return -1;
}
```

```
0 3 7
4 5 7
4 4 4
4
```

What will be printed when this code executes?

```
int[] values = { 0, 26, 55, 82, 91, 104, 112, 124 };

System.out.println(search(values, 91));
```

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	??	??

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	??	??

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	2	2

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	2	2
4	??	??

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	2	2
4	4	3

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	2	2
4	4	3
8	??	??

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	2	2
4	4	3
8	8	4

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	2	2
4	4	3
8	8	4
...	...	...
4,294,967,296	4,294,967,296	??

How Fast Is Binary Search?

Number of array elements we need to examine in the worst case:

Array Size	Sequential Search	Binary Search
1	1	1
2	2	2
4	4	3
8	8	4
...	...	...
4,294,967,296	4,294,967,296	33

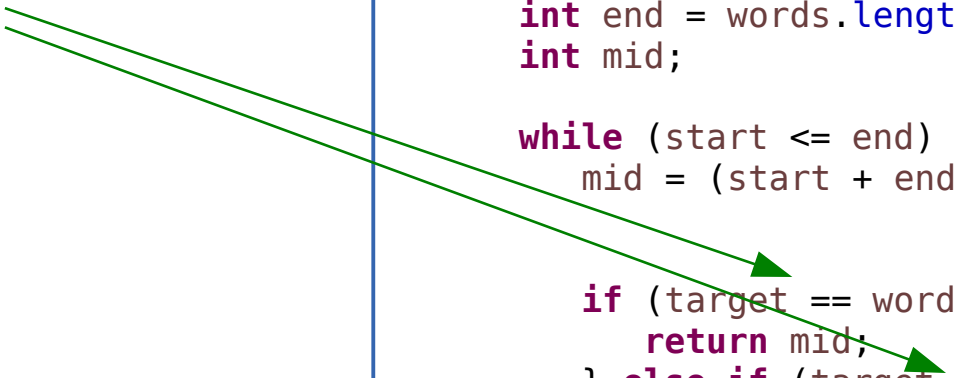
What About Strings?

- Is this OK?

```
public static int search(String[] words, String target) {

   int start = 0;
   int end = words.length - 1;
   int mid;

   while (start <= end) {
      mid = (start + end) / 2;


      if (target == words[mid]) {
         return mid;
      } else if (target < words[mid]) {
         end = mid - 1;
      } else {
         start = mid + 1;
      }
   }

   return -1;

}
```

What About Strings?

- Is this OK?
- NO!



```
public static int search(String[] words, String target) {

   int start = 0;
   int end = words.length - 1;
   int mid;

   while (start <= end) {
      mid = (start + end) / 2;


      if (target == words[mid]) {
         return mid;
      } else if (target < words[mid]) {
         end = mid - 1;
      } else {
         start = mid + 1;
      }
   }

   return -1;

}
```

String Comparisons

- You know all about .equals:

```
a.equals(b)
```

- We also have .compareTo:

```
a.compareTo(b)  // returns -1 if “a < b”
                // returns 0 if “a == b”
                // returns 1 if “a > b”
```

- Typical usage:

```
if (a.compareTo(b) < 0) {
    System.out.println("a is less than b!");
}
```

Correct Binary Search With Strings

```
public static int search(String[] words, String target) {

   int start = 0;
   int end = words.length - 1;
   int mid;

   while (start <= end) {
      mid = (start + end) / 2;

      if (target.equals(words[mid])) {
         return mid;
      } else if (target.compareTo(words[mid]) < 0) {
         end = mid - 1;
      } else {
         start = mid + 1;
      }
   }

   return -1;
}
```