

Ch 16.2 Search

Search = find a specific item in a collection of items

Sequential search

Loop sequentially through an array or list.

Performance searching a collection with N items:

- Average performance = $N/2$
- Worst performance = N // not finding the item

Binary search

IMPORTANT - Requires items to be pre-sorted/pre-processed.

Pseudo-code:

```
// search array A for KEY, return index if found
first = 0
last = (num items in A) - 1
while first <= last {
    middle = (last + first)/2;
    if A[middle] == KEY
        return middle // found!
    else if (A[middle] < KEY)
        first = middle + 1 // now search upper sub-array
    else
        last = middle - 1; // now search lower sub-array
}
return not found // not found if we exit loop
```

There is a recursive binary search as well. See page 986.

Performance searching a collection of N items:

- Average performance = $\log N$
- Worst performance = $\log N$ // not finding item

Quiz: For 1,000,000 items, performance of sequential may be 1M searches. What is it for binary search?

This is all well and good to search for integers.. what about searching for objects?

Comparable interface

The `Comparable` interface defines one method:

```
int compareTo( T obj)
```

To use this method, it's two steps:

1. Your class must implement `Comparable`, and then
2. You must write the `compareTo()` method for your class

Comparator objects

The `Comparator` interface defines two methods:

```
int compare( T o1, T o2)  
boolean equals(Object obj)
```

How do you choose between these two approaches? It's usually pretty easy.

- Fast and dirty, but less flexible - `Comparable` because you just add a method, but then your search/sort is pretty much limited to one criteria
- More complicated, but more powerful - `Comparator` because you have to create a separate (private?) class, but once you do, your options are pretty much unlimited

Example usages are found in Java classes `Arrays` and `Collections`. These are helper classes where nearly all of the methods are static.

From the `Arrays` class (docs.oracle.com/javase/7/docs/api/java/util/Arrays.html):

- `public static int binarySearch(Object[] a, Object key)`
- `static int binarySearch(T[] a, T key, Comparator<T> c)`

From the `Collections` class (docs.oracle.com/javase/7/docs/api/java/util/Collections.html):

- `static int binarySearch(List<T> list, T key)`
- `static int binarySearch(List<T> list, T key, Comparator<T> c)`

Quiz: When comparing two objects, what is the difference between `compareTo()` and `"=="`?