

Search and Sort

Michael Wagner

Bubble Sort

```
public static void bubbleSort(String[] haystack)
{
    boolean swapped;

    do
    {
        swapped = false;
        for (int subscript = 0; subscript < haystack.length - 1; subscript++)
        {
            int compare = haystack[subscript + 1].compareTo(haystack[subscript]);
            if (compare < 0)
            {
                String temp = haystack[subscript];
                haystack[subscript] = haystack[subscript + 1];
                haystack[subscript + 1] = temp;
                swapped = true;
            }
        }
    } while (swapped);
}
```

Bubble Sort

```
public static void main(String[] args)
{
    String[] haystack = {"China", "Japan", "Korea", "Norway", "England", "Germany"};

    bubbleSort(haystack);

    for (String current : haystack)
    {
        System.out.println(current);
    }
}
```

Example

Using the debugger, step through the Bubble Sort algorithm (code on course site).

Selection Sort

```
public static void selectionSort(String[] haystack)
{
    for (int change = 0; change < haystack.length; change++)
    {
        int least_subscript = change;
        for (int subscript = change + 1; subscript < haystack.length; subscript++)
        {
            int compare = haystack[subscript].compareTo(haystack[least_subscript]);
            if (compare < 0)
            {
                least_subscript = subscript;
            }
        }
        String temp = haystack[change];
        haystack[change] = haystack[least_subscript];
        haystack[least_subscript] = temp;
    }
}
```

Selection Sort

```
public static void main(String[] args)
{
    String[] haystack = {"China", "Japan", "Korea", "Norway", "England", "Germany"};

    selectionSort(haystack);

    for (String current : haystack)
    {
        System.out.println(current);
    }
}
```

Example

Using the debugger, step through the Selection Sort algorithm (code on course site).

Linear Search

```
public class Search
{
    public static int indexOf(String[] haystack, String needle)
    {
        for (int subscript = 0; subscript < haystack.length; subscript++)
        {
            if (needle.equals(haystack[subscript]))
            {
                return subscript;
            }
        }
        return -1;
    }

    public static void main(String[] args)
    {
        String[] haystack = {"China", "Japan", "Korea", "Norway", "England", "Germany"};

        int result_found = indexOf(haystack, "Norway");
        int result_not_found = indexOf(haystack, "Brazil");

        System.out.println(result_found);
        System.out.println(result_not_found);
    }
}
```


Binary Search

```
public static int indexOf(String[] haystack, String needle)
{
    int lower = 0;
    int upper = haystack.length;

    while (lower <= upper)
    {
        int subscript = (lower + upper) / 2;
        int compare = needle.compareTo(haystack[subscript]);
        if (compare < 0)
        {
            upper = subscript - 1;
        }
        else if (compare > 0)
        {
            lower = subscript + 1;
        }
        else
        {
            return subscript;
        }
    }
    return -1;
}
```

Binary Search - Limitation

```
public static void main(String[] args)
{
    String[] haystack = {"China", "Japan", "Korea", "Norway", "England", "Germany"};

    Arrays.sort(haystack);

    int result_found = indexOf(haystack, "Norway");
    int result_not_found = indexOf(haystack, "Brazil");

    System.out.println(result_found);
    System.out.println(result_not_found);
}
```