Sequential Search

Looking for something?

- Method for finding an item in a list
- Search checks each element in the array until the item is found OR the entire array is searched
- AKA linear search

An example

- The method (seqSearch) receives two variables called *list* which is an array of strings and *item* of type string. This method performs a sequential search that locates item in list.
    - If the search is successful, the method returns the index in the array of item but, if the search fails, the method returns -1.

Example 1 - Trace

```
public static int seqSearch (String[] list, String item)
{
        int location = -1;
        for (int i = 0; i < list.length; i++)
        if (list[i] .equals(item))
                location = i;
        return location;
}


public static void main(String[] args) {
{
        String [] x = {"I","love","CS","a","lot"};
        System.out.println(seqSearch(x,"CS"));
        System.out.println(seqSearch(x,"Hi"));

}
```

Efficiency

- Even if it finds item early in the search, it continues to look through the rest of the array

- We can correct this defect by using a boolean variable that acts as a flag to stop the search as soon as item has been found.

Example 2:

```
public static int seqSearch (String[] list, String item)
{
        int location = -1;
        boolean found = false;
        for (int i = 0; i < list.length && !found; i++)
                if (list[i] .equals (item))
                {
                        location = i;
                        found = true;
                }
        return location;
}
```

Example 3:

```
public static int seqSearch (String[] list, String item)
{
    for (int i = 0; i < list.length; i++)
        if (list[i] .equals(item))
            return i;
    return -1;
}
```