Reading:
  1. Dale, Chapter 11 (Section 6)

Objectives:
  1. Finish discussion on searching

Concepts:
  1. Searching
1. Discuss searching
   - Discuss searching and its performance with sorted and unsorted lists
   - Discuss different searching algorithms and their $O(n)$ (big-O):
     - Binary search
     - Sequential search
     - Sorted array search
     - Hashing
Objective:
1. Practice binary search use

Student Activities:
1. Use the binary search function that was provided to you last week and fill it in the program provided below. Then execute the program and make sure that the binary search is working the way it is supposed to. Once you do that perform a binary search for the following: 25, 8, 6. Show the output you got when searching for each one.

```java
import java.awt.*;
import java.applet.Applet;

public class BinarySearch extends Applet {
    int a[];
    int element;
    String searchKey;
    int xPosition;   // applet horizontal drawing position
    int yPosition;   // applet vertical drawing position
    Label enterLabel;
    TextField enter;
    Label resultLabel;
    TextField result;
    Boolean timeToSearch = false;

    public void init()
    {
        a = new int[15];
        for ( int j = 0; j <a.length; j++)  // create data
            a[j] = 2 * j;

        enterLabel = new Label ( "Enter key" );
        enter = new TextField(5);
        resultLabel = new Label ("Result");
        result = new TextField(22);
        result.setEditable(false);
        add( enterLabel );
        add( enter );
        add( resultLabel );
        add( result );
    }

    public void paint ( Graphics g )
```
if (timeToSearch) {  // prevents search 1st time called
    element = binarySearch(Integer.parseInt(searchKey), g);

    if (element != -1)
        result.setText("Found value in element "+ element);
    else
        result.setText("Value not found");
}

public boolean action (Event event, Object o)
{
    if (event.target == enter) {
        timeToSearch = true;
        xPosition = 25;
        yPosition = 75;
        searchKey = event.arg.toString();
        repaint();  // call paint to start search and output
    }
    return true;
}

public int binarySearch (int key, graphics gg)
{
    // FILL in the code
}

// Print one row of output showing the current part of the array being processed
void printRow (int low, int mid, int high, Graphics gg)
{
    xPosition = 25;
    for (int j=0; j < a.length; j++) {
        if (j < low || j > high)
            gg.drawstring("", xPosition, yPosition);
        else if (j == mid)
            gg.drawstring(String.valueOf(a[j]) + "+", xPosition, yPosition);
        else
            gg.drawstring(String.valueOf(a[j]), xPosition, yPosition);
        xPosition = 20;
    }
    yPosition += 15;
}
The correct binary search code was provided last week. The output for the numbers 25, 8, 6 is supposed to be:

Enter key 25   Result **Value not found**

Portions of the array searched
0 2 4 6 8 10 12 14* 16 18 20 22 24 26 28
16 18 20 22* 24 26 28
24 26* 28
24*

Enter key 8   Result **Found value in element 4**

Portions of the array searched
0 2 4 6 8 10 12 14* 16 18 20 22 24 26 28
0 2 4 6* 8 10 12
8 10* 12
8*

Enter key 6   Result **Found value in element 3**

Portions of the array searched
0 2 4 6 8 10 12 14* 16 18 20 22 24 26 28
0 2 4 6* 8 10 12