AP Computer Science A
Ticket Out the Door
Set 35: Sorts and Searches

Name _____ Period _____

Skill 35.1 Exercise 1

Consider the following list,

```
var arr = [1, 3, 5, 6, 0, -1, 2, 10, 4];
```

Write a function call `minToFront`, that accepts a parameter called `front`, which represents the front of the list. In the body of the function, write an algorithm that locates the lowest value in the list and swaps it with the `front` location.

Now that we have an algorithm that moves the lowest item in a list to the front, write another function below called `selectionSort`, that sorts the list from low to high.

If an array contains the following elements, what would the array look like after the third pass of `selectionSort`, sorting from high to low?

89 42 -3 13 109 70 2

AP Computer Science A
Ticket Out the Door
Set 35: Sorts and Searches

Name _____ Period _____

Skill 35.2 Exercise 1

Write the function `search`, that can be implemented as illustrated below,

```
    var arr[] = {2, 3, 4, 10, 40};
    var x = 10;
    var result = search(arr, x);
    if(result == -1)
        console.log("Element is not present in array");
    else
        console("Element is present at index " + result);


//complete the search function below.
```

Under what conditions is this type of search impractical?

AP Computer Science A
Ticket Out the Door
Set 35: Sorts and Searches

Name _____ Period _____

Skill 35.3 Exercise 1

For which of the following arrays could a binary search be applied? Explain.

{1, 10, 22, 32, 100, 200, 302}

{x, y, z, a, b, c, d, f}

{and, ant, bat, cat, dog, rat}

{300.12, 200, 100, 50, 2, 0, -80}

Skill 35.3 Exercise 2

Consider the following binarySearch function. The function correctly performs a binary search.

```
/** Precondition: data is sorted in increasing order. */

function binarySearch(data, target) {

var start = 0;
var end = data.length - 1;

while (start <= end)   {
     var mid = Math.floor((start + end) / 2);     /* Calculate midpoint */

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

Consider the following code segment.

```
var values = [1, 2, 3, 4, 5, 8, 8, 8];
var target = 8;
```

What value is returned by the call binarySearch(values, target) ?

Suppose the binarySearch method is called with an array containing 2,000 elements sorted in increasing order.

What is the maximum number of times that the statement indicated by /* Calculate midpoint */ could execute?

AP Computer Science A
Ticket Out the Door
Set 35: Sorts and Searches

Name _____ Period _____