

## 1) (10 pts) DSN (Recursive Coding)

Write an *efficient recursive* function that takes in a sorted array numbers, two integers, low and high, representing indexes into the array, and another integer, value, and returns the index in the array where value is found in the array in between index low and high, inclusive. If value is NOT found in the array in between indexes low and high, inclusive, then the function should return -1.

```
int search(int numbers[], int low, int high, int value) {  
  
    if (low > high) return -1;  
  
    int mid = (low+high)/2;  
  
    if (value > numbers[mid])  
        return search(numbers, mid+1, high, value);  
    else if (value < numbers[mid])  
        return search(numbers, low, mid-1, value);  
    else  
        return mid;  
}
```

**Grading:** 2 pts for return -1 base case

3 pts for going halfway in between search range

2 pts for case going to the right

2 pts for case going to the left

1 pt for base case returning mid

Max grade of 7 for linear recursive solution (no pts for going halfway...)

Max grade of 3 pts for non-recursive solution, regardless of runtime.

Max grade of 6 if recursive structure is correct but recursive call(s) are missing