

Binary Search Code
AP Computer Science A
Mr. Fahrenbacher

Recursive Binary Search

```
public int binarySearch(int[] array, int target, int lo, int hi) {
    if(lo > hi)
        return -1;
    int mid = (lo + hi) / 2;

    if(array[mid] < target) //search right
        return binarySearch(array, target, mid+1, hi);
    else if(array[mid] > target) //search left
        return binarySearch(array, target, lo, mid-1);
    else //they must be equal!
        return mid;
}
```

or

Looped Binary Search

```
public int binarySearch(int[] array) {
    int lo = 0;
    int hi = array.length-1;
    int mid;

    while(lo <= hi) { //while our lo and hi values make sense
        mid = (lo + hi) / 2;

        if(array[mid] < target) //search right
            lo = mid+1;
        else if(array[mid] > target) //search left
            hi = mid - 1;
        else //ew found it
            return mid;
    }

    return -1; //we didn't find it
}
```

Mr. Fahrenbacher
 Binary Search Practice
 AP Computer Science A

Example:

0	1	2	3	4	5	6	7	8	9
-90	-89	-50	-2	1	13	14	17	23	104

Target: 2

Stage 1: lo = 0 hi = 9 mid = (hi+lo)/2 = 4
 Middle value (1) is less than target (2), so search right

Stage 2: lo = mid + 1 = 5 hi = 9 mid = 7
 Middle value (17) is greater than target (2), so search left

Stage 3: lo = 5 hi = mid - 1 = 6 mid = 5
 Middle value (13) is greater than target (2), so search left

Stage 4: lo = 5 hi = mid - 1 = 4 STOP!
 Hi index (4) is less than lo index (5), so target value (2) cannot be found

Directions: Practice binary search on the following arrays. You only need to keep track of the low, hi, and mid indexes - I added the extra english for instructive purposes.

1. Target Value: 4

0	1	2	3	4	5	6	7	8	9
-2	-1	0	1	2	4	8	16	32	64

2. Target Value: 5

0	1	2	3	4	5	6	7	8
5	6	10	13	24	42	81	82	83

3. Target Value: 32

0	1	2	3	4	5	6	7	8	9
-2	-1	0	1	2	4	8	16	32	64

4. Target Value: 14

0	1	2	3	4	5	6	7	8
5	6	10	13	24	42	81	82	83