**Algorithm HW3**

**Question: We want to find the cutting position in case of an ascending array.**

Assume an array has n elements. The input array just would be two sorted part and both are in ascending order.

At most take n operations to search the cutting point.

EX:

Input arr[]: 3, 4, 5, 6, 7, 1

^ ^

Sorted array[]: 1, 3, 4, 5, 6, 7 return 0

Method 1:

We traverse the input array from both ends for each operation. Then, it can guarantee we can find the turning point by at most n/2 operations.

int left, right;

int found // `found` represents we already find the turning point.

Method 2:

Use binary search to find the turning point.

EX:

Input arr[]: 3, 4, 5, 6, 7, 1

^

5, 6, 7, 1, 2, 3, 4

^

maxElemIndex = 6

Low = 2

High = 2

Mid =

// ` orderingFlag` = 0 : ascending

// ` orderingFlag` = 1 : descending

orderingFlag = 0

int low = 0

int high = 5

while (low < high) {

int mid = (low + high) / 2 = 2

if (orderingFlag == 0) {

if (arr[mid] >= arr[low]) {

low = mid;

} else {

high = mid;

orderingFlag = 1;

}

}

else if (orderingFlag == 1) {

if (arr[mid] <= arr[high]) {

high = mid – 1;

} else {

Low = mid

}

}

}