Algorithms

1. **Majority Voting Algorithm**

## Basic sort and search:

Sort the input array and search for max frequency item.

## Boyer Moore algorithm:

Take a candidate and count its frequency. See the algorithm below:

candidate = 0

count = 0

for value in input:

if count == 0:

candidate = value

if candidate == value:

count += 1

else:

count -= 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Space complexity | Time Complexity |  |  |
| Basic sort and Search | O(1) | O(n\*logn) |  |  |
| boyer moore algorithm | O (1) | O(n) |  |  |

References: <https://gregable.com/2013/10/majority-vote-algorithm-find-majority.html>

1. **Count Number of Occurrences**

## Linear Search:

Search the array linearly and count the number of times k is repeated in the array.

## Binary Search:

Find the first occurrence(i) of k in array and last occurrence(j) of k in array. Return (j-i+1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Space complexity | Time Complexity |  |  |
| Linear Search | O(1) | O(n) |  |  |
| Binary search | O(1) | O(log n) |  |  |

References: 1. <http://www.geeksforgeeks.org/count-number-of-occurrences-or-frequency-in-a-sorted-array/>