

Binary Search Algorithm

Task Requirements:

Write an application that reads binary array, then find the value from the binary array using key, using Binary search Algorithm.

Binary Search Algorithm, finds the position of a target value within a sorted array. Binary search compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value, and repeating this until the target value is found. If the search ends with the remaining half being empty, the target is not in the array.

```
public static int binarySearch(int[] inputArray, int key) {  
    int start = 0;  
    int end = inputArray.length - 1;  
    Arrays.sort(inputArray);  
    while (start <= end) {  
        int middle = (start + end) / 2;  
        if (key == inputArray[middle]) {  
            return middle;  
        }  
        if (key < inputArray[middle]) {  
            end = middle - 1;  
        }  
        else {  
            start = middle + 1;  
        }  
    }  
    return -1;  
}
```

Binary Search Algorithm divide the list into two parts, and compares for key. Until start of the list \leq end of the list.

- If key is found in the middle of list return middle.
- If the item in the middle of the list is greater than our key, we should look for the key in the bottom half of the list, so we calculate a new value for end.
- If the item in the middle of the list is less than our key, we should look for the key in the top half of the list, so we calculate a new value for start.
- If the key is not in the list return false.