Problem Challenge 1

We'll cover the following

Search Bitonic Array (medium)

Given a Bitonic array, find if a given ‘key’ is present in it. An array is considered bitonic if it is monotonically increasing and then monotonically decreasing. Monotonically increasing or decreasing means that for any index `i` in the array `arr[i] != arr[i+1]`.

Write a function to return the index of the ‘key’. If the ‘key’ is not present, return -1.

Example 1:

```
Input: [1, 3, 8, 4, 3], key=4
Output: 3
```

Example 2:

```
Input: [3, 8, 3, 1], key=8
Output: 1
```

Example 3:

```
Input: [1, 3, 8, 12], key=12
Output: 3
```

Example 4:

```
Input: [10, 9, 8], key=10
Output: 0
```

Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```
def search_bitonic_array(arr, key):
  # TODO: Write your code here
  return -1

def main():
  print(search_bitonic_array([1, 3, 8, 4, 3], 4))
  print(search_bitonic_array([3, 8, 3, 1], 8))
  print(search_bitonic_array([1, 3, 8, 12], 12))
  print(search_bitonic_array([10, 9, 8], 10))


main()
```

Run | Save | Reset

← Back

Bitonic Array Maximum (easy)

Next →

Solution Review: Problem Challenge 1

☑ Completed

Report an Issue

Ask a Question