
1. Basic Binary Search

Question: You are given a sorted array. Find the target element using binary search.

Approach to Solve:

- Initialize `start = 0` and `end = n-1`.
 - Calculate `mid = (start + end) / 2`.
 - Compare the target with the element at `mid`.
 - If the target matches, return `mid`.
 - If the target is smaller, search the left half (`end = mid - 1`).
 - If the target is larger, search the right half (`start = mid + 1`).
 - Continue until you find the target or the search range becomes invalid (`start > end`).
-

2. First and Last Position of an Element

Question: Find the first and last occurrence of a target element in a sorted array.

Approach to Solve:

- Use binary search to find the **first occurrence**:
 - When the target is found, move the `end` pointer to `mid - 1` to continue searching the left half.
 - Use binary search to find the **last occurrence**:
 - When the target is found, move the `start` pointer to `mid + 1` to continue searching the right half.
 - If the target is not found, return `-1`.
-

3. Finding Peak Element

Question: Find a peak element in an array where an element is greater than or equal to its neighbors.

Approach to Solve:

- Calculate `mid = (start + end) / 2`.
 - Check if `arr[mid]` is greater than or equal to both neighbors:
 - If true, return `mid` as the peak element.
 - If the left neighbor is greater, search the left half (`end = mid - 1`).
 - If the right neighbor is greater, search the right half (`start = mid + 1`).
 - Continue until a peak element is found.
-

4. Count Occurrences of a Target

Question: Count the occurrences of a target element in a sorted array.

Approach to Solve:

- Use binary search to find the **first occurrence** of the target.
 - Use binary search to find the **last occurrence** of the target.
 - If the target is found, calculate the number of occurrences as $\text{last} - \text{first} + 1$.
 - If the target is not found, return 0.
-

5. Find the Element in a Rotated Sorted Array

Question: Find the index of a target element in a rotated sorted array.

Approach to Solve:

- Calculate $\text{mid} = (\text{start} + \text{end}) / 2$.
 - Check if the left part of the array is sorted:
 - If the target lies within this range, search the left half ($\text{end} = \text{mid} - 1$).
 - Otherwise, search the right half ($\text{start} = \text{mid} + 1$).
 - If the right part is sorted, adjust the search accordingly.
 - Continue adjusting the search bounds until the target is found or the array is exhausted.
-

These are the questions along with the approach to think through and solve them.