

3.12 FINDING THE K-th SMALLEST ELEMENT USING MEDIAN OF MEDIANS

Question:

To Implement a function `median_of_medians(arr, k)` that takes an unsorted array `arr` and an integer `k`, and returns the `k`-th smallest element in the array.

AIM

To implement a function `median_of_medians(arr, k)` that returns the `k`-th smallest element in an unsorted array `arr` using the Median of Medians algorithm, ensuring worst-case linear time complexity.

ALGORITHM

1. Divide the array into groups of 5 elements.
2. Sort each group and find the median.
3. Recursively apply the algorithm to find the median of medians.
4. Use this median as a pivot to partition the array into:
5. Elements less than pivot
6. Elements equal to pivot
7. Elements greater than pivot
8. Recurse into the appropriate partition based on the value of `k`.

PROGRAM

```
def partition(arr, pivot):
    lows = [el for el in arr if el < pivot]
    highs = [el for el in arr if el > pivot]
    pivots = [el for el in arr if el == pivot]
    return lows, pivots, highs

def select(arr, k):
    if len(arr) <= 5:
        return sorted(arr)[k - 1]

    medians = [sorted(arr[i:i+5])[len(arr[i:i+5])//2] for i in range(0, len(arr), 5)]
    pivot = select(medians, len(medians)//2 + 1)
    lows, pivots, highs = partition(arr, pivot)

    if k <= len(lows):
        return select(lows, k)
    elif k <= len(lows) + len(pivots):
        return pivot
    else:
        return select(highs, k - len(lows) - len(pivots))

def median_of_medians(arr, k):
    return select(arr, k)

def run_median_input():
    arr = list(map(int, input("Enter array elements separated by space: ").split()))
    k = int(input("Enter k (1-based index): "))
    print("K-th smallest element:", median_of_medians(arr, k))

run_median_input()
```

Input:

[65,12,87,56,93,17,6,90,4], key=6

Output:

```
>>> Enter array elements separated by space: 65 12 87 56 93 17 6 90 04
      Enter k (1-based index): 6
      K-th smallest element: 65
>>> |
```

RESULT:

Thus search program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- **Time Complexity:** $O(n)$
- **Space Complexity:** $O(n)$