Q Valid Partitions

Given the array `[4, 5, 1, 3, 7, 2, 6]`, which arrays are valid partitions on the pivot 4?

`[4, 5, 1, 3, 7, 2, 6]`

`[1, 2, 3, 4, 5, 6, 7]`

`[1, 7, 5, 3, 4, 2, 6]`

`[1, 3, 4, 2, 5, 7, 6]`

`[1, 3, 4, 2, 5, 6, 7]`

`[2, 1, 3, 4, 5, 7, 6]`

`[3, 2, 1, 4, 7, 5, 7]`

2

Given the array [4, 5, 1, 3, 7, 2, 6], which arrays are valid partitions on the pivot 4?

Q1: [4, 5, 1, 3, 7, 2, 6]

Q2: [1, 2, 3, 4, 5, 6, 7]

Q3: [1, 7, 5, 3, 4, 2, 6]

Q4: [1, 3, 4, 2, 5, 7, 6]

Q5: [1, 3, 4, 2, 5, 6, 7]

Q6: [2, 1, 3, 4, 5, 7, 6]

Q7: [3, 2, 1, 4, 7, 5, 7]

Q Hoare Partitioning

Given the array `[4, 5, 1, 3, 7, 2, 6]` and pivot 6, give the array after the first swap. Then, give the array after the entire partition operation is complete.

Is Hoare partitioning stable?

4

Q1: Given the array [4, 5, 1, 3, 7, 2, 6] and pivot 6, give the array after the first swap.

Q2: Then, give the array after the entire partition operation is complete.

Q3: Is Hoare partitioning stable?

Q Selection Algorithms

Selection. Given an array of N items, find item of rank **K**.

For finding the median, choose **K** = N / 2.

Characterize the difficulty of this problem.

- Why is the time complexity of selection in Ω(N)?
- Describe an O(N log N) runtime algorithm for selection for any **K**.
- Describe an O(N) runtime algorithm for selection for **K** = 0, 1, 2.

Q1: Why is the time complexity of selection in Ω(N)?

Q2: Describe an O(N log N) runtime algorithm for selection for any K.

Q3: Describe an O(N) runtime algorithm for selection for K = 0, 1, 2.

Q Partition Sort

Consider a generalization of quicksort: `PartitionSort(input, lo, hi, subsort)`.

1. If `hi - lo ≤ 1`, return.
2. Partition around `input[lo]`, the leftmost item of the current subproblem.
3. Call `subsort` on the left and right subproblems.

Give the worst-case runtime for each call below.

```
PartitionSort(input, 0, N, InsertionSort)
PartitionSort(input, 0, N, MergeSort)
PartitionSort(input, 0, N, PartitionSort)
```


Give the worst-case runtime for each call below.

Q1: PartitionSort(input, 0, N, InsertionSort)

Q2: PartitionSort(input, 0, N, MergeSort)

Q3: PartitionSort(input, 0, N, PartitionSort)