

Mr. Y has learned about quicksort recently. He knows the worst-case time complexity of quicksort is  $O(n^2)$ . He also knows that the worst-case scenario can easily be avoided by using "Randomized Pivoting". He searched about it on the internet and found the following solution in GeeksforGeeks.

```
partition(arr[], lo, hi)
    pivot = arr[hi]
    i = lo    // place for swapping
    for j := lo to hi - 1 do
        if arr[j] <= pivot then
            swap arr[i] with arr[j]
            i = i + 1
    swap arr[i] with arr[hi]
    return i

partition_r(arr[], lo, hi)
    r = Random Number from lo to hi
    Swap arr[r] and arr[hi]
    return partition(arr, lo, hi)

quicksort(arr[], lo, hi)
    if lo < hi
        p = partition_r(arr, lo, hi)
        quicksort(arr, p-1, hi)
        quicksort(arr, p+1, hi)
```

Mr. Y is a curious young mind. He wants to do something new. So he found out a new approach for randomized quicksort which is different from the above one. Be like Mr. Y.