

In lab exercise

The midterm claimed the function  $\text{Center}(A)$  could run in  $O(n)$  time. Let's look at the details of such an algorithm. Assume  $\text{Center}$  takes an unsorted array  $A$  of size  $n$  as input. We wish to find the median value of  $A$ . Actually, in practice, we will write  $\text{Center}$  as  $\text{Rank}(A, \lfloor n/2 \rfloor)$ . We describe  $\text{Rank}(A, i)$  below.

1. Divide the  $n$  elements into  $\lfloor n/5 \rfloor$  groups of 5 elements each and at most one group of  $n \bmod 5$  elements.
2. Find the median of each of the groups of size 5 =  $O(1)$
3. Use the  $\text{Rank}$  algorithm to recursively find the median  $x$  of the  $\lfloor n/5 \rfloor$  medians from the previous step.
4. Partition  $A$  around  $x$ . Assume  $x$  is the  $k$ th item after partitioning, meaning there are  $n - k$  elements larger than  $x$  and  $k$  elements smaller than or equal to  $x$ .
5. If  $k = i$ , return  $x$ . Otherwise run  $\text{Rank}$  recursively to find the  $i$ th smallest element on the low side of the partition if  $i < k$  or the  $(i - k)$ th smallest element on the high side if  $i > k$ .

Does this algorithm work? Explain. Write and solve a recurrence to analyze its runtime. Why do we partition the array into groups of 5? Would other values work? Would some group sizes not work?