Theorem 1. The randomized kth algorithm runs in \( O(n) \) expected time.

Proof. As in class, let \( T(n, k) \) be the expected running time of the \( k \)th algorithm on an array of size \( n \) and with argument \( k \). Let \( T(n) = \max_k T(n, k) \). Then \( T(1) = 1 \).

The partition phase of the algorithm may split the array into two sub-arrays of size \( t \) and \( n - t \), where each \( t \) is equally likely. Which array gets used in the recursive call depends on \( k \). To handle this, let’s assume that the algorithm always recurses on the larger of the two sub-arrays. If we can prove such an algorithm runs in \( O(n) \) time, then the original algorithm must also run in \( O(n) \) time.

The larger of the two halves will always have size at least \( n/2 \). For each size \( s \geq n/2 \), there are two possible values of \( t \) (from the partition call) that can result in a sub-array of size \( s \):

\[ t = s \]

\[ t = n - s \]

Thus,

\[ \Pr[\text{recurse on sub-array of size } s] \leq \frac{2}{n} \]

This gives us the following recurrence relation on the expected running time:

\[ T(n) \leq \sum_{s=n/2}^{n-1} \frac{2}{n} T(s) + cn \]

where \( cn \) is a bound on the running time of the partition algorithm.

We now prove by induction that \( T(n) \leq 4cn \). This is trivially true for \( n = 1 \). So suppose that for all \( n < N \), \( T(n) \leq 4cn \). Then

\[ T(N) \leq \sum_{s=N/2}^{N-1} \frac{2}{N} T(s) + cN \]

\[ = \frac{8c}{N} \sum_{s=N/2}^{N-1} s + cN \]

\[ = \frac{8c}{N} \left( \frac{N(N-1)}{2} - \frac{N}{2} \right) + cN \]

\[ \leq \frac{4c}{N} \left( N^2 - N - \frac{N^2}{4} + N + N \right) + cN \]

\[ \leq 3cN + cN \leq 4cN \]
Thus, by induction, $T(n) \leq 4cn$ for all $n \geq 1$. Thus, the expected running time of the randomized $k$th algorithm is $O(n)$.

The above proof also makes clear one of the questions we had in class. Recall that, during the analysis of the deterministic $k$th algorithm, we wanted to solve for a $c'$ such that the running time, $T(n)$, of the algorithm is less than $c'n$. We had the recurrence relation

$$T(n) \leq T\left(\frac{7n}{10}\right) + T\left(\frac{n}{5}\right) + cn$$

The question was: do we want to solve for $c'$ such that

$$c'n \leq \left(\frac{7c'n}{10}\right) + \frac{c'n}{5} + cn$$

or

$$c'n \geq \left(\frac{7c'n}{10}\right) + \frac{c'n}{5} + cn$$

The larger context – that this will be part of a proof by induction – makes it clear that we need to find $c'$ that makes the second inequality hold. Then we will be able to complete the following proof:

**Theorem 2.** The deterministic $k$th algorithm runs in $O(n)$ time.

**Proof.** We have that $T(1) = 1$ and

$$T(n) \leq T\left(\frac{7n}{10}\right) + T\left(\frac{n}{5}\right) + cn$$

where $cn$ is a bound on the running time of the partition algorithm. Let $c' = 10c$. We will prove by induction that $T(n) < c'n$ for all $n \geq 1$. This is trivially true for $n = 1$. Now suppose we’ve proven this fact for all $n < N$. Then

$$T(N) \leq T\left(\frac{7N}{10}\right) + T\left(\frac{N}{5}\right) + cN$$

$$\leq \frac{7c'N}{10} + \frac{c'N}{5} + cN$$

$$= \frac{9c'N}{10} + \frac{c'N}{5}$$

$$= \frac{c'N}{2}$$

Thus, by induction, $T(n) \leq 10cn$ for all $n \geq 1$. Consequently, the running time of the deterministic kth algorithm is $O(n)$.