1. Assuming all permutations of \((2n + 1)\) distinct numbers to be equally probable, what is the probability that the Lomuto Partition algorithm will produce two segments, each of size \(n\)?

2. Assuming all permutations of \((2n + 1)\) distinct numbers to be equally probable, what is the probability that the Lomuto Partition algorithm will produce two segments, one of size \(2n\) and the other of size 0?

3. We proved that all comparison-based sorts are \(\Omega(n \log n)\) in the worst case. Yet QuickSort is a comparison-based sort for which the worst case is \(\Theta(n^2)\), which implies \(\Omega(n^2)\). Resolve this apparent contradiction.

4. Suppose we run QuickSort on a permutation such that every level produces subproblems of sizes \((m/4) - 1\) and \((3m/4)\), where \(m\) is the size of the level being partitioned. Assume further that a partition exists for which all of these sizes are integers. Let level 0 be the initial problem of size \(n\). Let level 1 be the two subproblems of sizes \((n/4) - 1\) and \((3n/4)\), and so forth. Let \(k\) denote that lowest level. Find an expression for \(k\) in terms of \(n\).
Solutions

1. Assuming all permutations of \((2n + 1)\) distinct numbers to be equally probable, what is the probability that the Lomuto Partition algorithm will produce two segments, each of size \(n\)?

\[
\frac{(2n)!}{(2n + 1)!} = \frac{1}{2n + 1}, \text{ because the median value must be the pivot and thus must appear in the last position.}
\]

2. Assuming all permutations of \((2n + 1)\) distinct numbers to be equally probable, what is the probability that the Lomuto Partition algorithm will produce two segments, one of size \(2n\) and the other of size 0?

\[
\frac{2(2n)!}{(2n + 1)!} = \frac{2}{2n + 1}, \text{ because the pivot must be the smallest or largest value.}
\]

3. We proved that all comparison-based sorts are \(\Omega(n \log n)\) in the worst case. Yet QuickSort is a comparison-based sort for which the worst case is \(\Theta(n^2)\), which implies \(\Omega(n^2)\). Resolve this apparent contradiction.

Let \(T(n)\) be the worst-case complexity of QuickSort. Then \(T(n) = \Theta(n^2)\) gives, for some \(K > 0\),

\[
T(n) \geq K \cdot n^2 \geq K \cdot n \log n.
\]

Therefore \(T(n) = \Omega(n \log n)\). There is no contradiction.

4. Suppose we run QuickSort on a permutation such that every level produces subproblems of sizes \((m/4) - 1\) and \((3m/4)\), where \(m\) is the size of the level being partitioned. Assume further that a partition exists for which all of these sizes are integers. Let level 0 be the initial problem of size \(n\). Let level 1 be the two subproblems of sizes \((n/4) - 1\) and \((3n/4)\), and so forth. Let \(k\) denote that lowest level. Find an expression for \(k\) in terms of \(n\).

As in the homework, we find the deepest level by following the decomposition of the larger part:

\[
n \rightarrow \left(\frac{3}{4}\right) n \rightarrow \left(\frac{3}{4}\right)^2 n \rightarrow \left(\frac{3}{4}\right)^3 n \rightarrow \ldots \rightarrow \left(\frac{3}{4}\right)^k n = 1
\]

\[
\left(\frac{3}{4}\right)^k = \frac{1}{n}
\]

\[
k \log \left(\frac{3}{4}\right) = \log \left(\frac{1}{n}\right) = - \log n
\]

\[
k = \frac{- \log n}{\log(3/4)} = \frac{\log n}{\log(4/3)}.
\]