QuickSort

Associated reading: §11.2, Goodrich/Tamassia text.

QuickSort proceeds as follows. First we're going to divide the array into two parts, hopefully halves. We then recursively sort each half. The key operation is `partition`, which is based on a *pivot*. Different flavors of QuickSort differ primarily on how they select the pivot, although different mechanisms for partition exist as well.

48	41	51	66	84	89	87	68	37	23	96	98	52	14	62	43

Question 1. Suppose I am going to sort a vector of n comparable objects by QuickSort. All $n!$ initial permutations are equally likely. What is the expected average case running time for the algorithm? For simplicity, assume all n keys are distinct.

Let $P_{i,j}$ be probability we compare S_i and S_j. What is $P_{i,j}$.

Let $X_{i,j}$ be an indicator random variable for whether or not i, j get compared. Because that is a binary outcome, $X_{i,j} = 1 \cdot P_{i,j} + 0 \cdot (1 - P_{i,j}) = P_{i,j}$

$$
\begin{aligned}
E\Big(\sum_{i=1}^{n}\sum_{j=i+1}^{n} X_{i,j}\Big) &= \sum_{i=1}^{n}\sum_{j=i+1}^{n} E(X_{i,j}) \\
&= \sum_{i=1}^{n}\sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n}\sum_{k=2}^{n-i+1} \frac{2}{k} \\
&< \sum_{i=1}^{n}\sum_{k=1}^{n} \frac{2}{k}
\end{aligned}
$$