RandomSelect is a randomized selection algorithm described in the class and in the text. It runs in expected \( O(n) \) time and worst-case \( O(n^2) \) time. You are asked to implement RandomSelect and examine its running time for various input sizes.

To examine the running times, generate a random sequence of numbers using the following algorithm:

\[
\text{RandomPermutation} \ (A[1..n]) \{ \\
\quad \text{for } j = 1 \text{ to } n-1 \{ \\
\qquad k = \text{Random} \ (j, n) \\
\qquad \text{swap} \ (A, j, k) \\
\}\}
\]

Here, Random \( j, n \) returns an integer in the range \( j \) through \( n \). So, if you initialize the array \( A \ [1..n] \) to be the first \( n \) natural numbers 1, 2, 3, ..., \( n \), and call RandomPermutation, then you get a random permutation of the first \( n \) natural numbers.

Run your RandomSelect algorithm for inputs of size 1000, 2000, 3000, 4000, 5000. Generate 10 inputs of each size and record the average running time of that size. Tabulate your results separately for each input type, plot the running times against the input sizes, and explain your results.