**Homework #10**

**Team 2:**

**Julian Carvajal Rico**

**Roberto Enriquez Vargas**

The performance metrics for parallel linear and binary searches as the number of threads increases provide insightful conclusions. For the linear search, there's a clear trend of improved speed with more threads, but with diminishing returns as the number of threads grows, especially evident when the efficiency drops from full efficiency with a single thread to about 61% with 16 threads. This suggests that while parallelization benefits linear search, the gains in speedup are mitigated by the overhead associated with managing an increasing number of threads.

In contrast, the binary search behaves differently in a parallel setting. The speedup for binary search not only fails to improve but worsens after increasing the thread count beyond one. Efficiency drops off sharply, highlighting that the binary search's divide and conquer nature might not be suitable for the parallelization strategy employed. The increase in execution time for binary search with more threads suggests that the overhead from parallelism overshadows any potential gains.

The contrast between the two search methods reflects an important aspect of parallel computing where not all algorithms benefit equally from an increase in computational resources. Linear search, while simpler, scales somewhat better with added threads up to a point, by the contrary the binary search demonstrates that some algorithms might perform best in a serial context due to their inherent efficiency and complexity of parallelization.

**Table 1. HW10 Results**

