CONCLUSION

1. **Brute force algorithm:** This algorithm checks every number up to 'n' to see if it is a prime by dividing it by all the integers less than it. This algorithm has a time complexity of O(n^2), which makes it impractical for large values of 'n'.

2. **Sieve of Eratosthenes:** This algorithm generates all prime numbers up to 'n' by marking all multiples of primes as non-prime. The time complexity of this algorithm is O(n*log(log(n))), which makes it very efficient for generating prime numbers.

3. **Trial division algorithm:** This algorithm generates all prime numbers up to 'n' by checking if each number is divisible by any integer less than or equal to the square root of the number. The time complexity of this algorithm is O(n*sqrt(n)), which is faster than the brute force algorithm but slower than the Sieve of Eratosthenes.

4. **Sieve of Sundaram:** This algorithm generates all prime numbers up to '2n+1' by removing all numbers of the form 'i+j+2ij' where i, j are positive integers such that 'i+j+2ij <= 2n'. The time complexity of this algorithm is O(n*log(n)), which is faster than the trial division algorithm but slower than the Sieve of Eratosthenes.

5. **Miller-Rabin primality test:** This probabilistic algorithm determines if a number 'n' is prime or composite by checking a number of bases for 'n'. The algorithm has a time complexity of O(k*log^3(n)) where 'k' is the number of bases checked, and it is very fast for large values of 'n'. However, it is not guaranteed to provide an accurate result for all numbers.

In summary, the Miller-Rabin primality test is the fastest algorithm for large values of 'n', with a time complexity of O(klog^3(n)). However, it is not guaranteed to provide an accurate result for all numbers. The Sieve of Eratosthenes is the most efficient deterministic algorithm for generating prime numbers with a time complexity of O(nlog(log(n))). The trial division algorithm and Sieve of Sundaram are less efficient, but they can still be useful for generating prime numbers when the value of 'n' is not too large. The brute force algorithm is the least efficient algorithm and should only be used for small values of 'n'.