Algorithm 3.1 A serial program for finding the minimum in an array of numbers \( A \) of length \( n \).
1. **procedure** RECURSIVE_MIN (A, n)
2. begin
3. if (n = 1) then
4.   min := A[0];
5. else
6.   lmin := RECURSIVE_MIN (A, n/2);
7.   rmin := RECURSIVE_MIN (&A[n/2], n − n/2);
8. if (lmin < rmin) then
9.   min := lmin;
10. else
11.   min := rmin;
12. endelse;
13. endelse;
14. return min;
15. end RECURSIVE_MIN

**Algorithm 3.2** A recursive program for finding the minimum in an array of numbers A of length n.
Algorithm 3.3 A serial column-based algorithm to factor a nonsingular matrix $A$ into a lower-triangular matrix $L$ and an upper-triangular matrix $U$. Matrices $L$ and $U$ share space with $A$. On Line 9, $A[i, j]$ on the left side of the assignment is equivalent to $L[i, j]$ if $i > j$; otherwise, it is equivalent to $U[i, j]$. 
Algorithm 3.4  A sample serial program to be parallelized.