

1. Write a greatest common divisor definition in Prolog. Define a relation $gcd(M, N, S)$ which is true if and only if the greatest common divisor of M and N is S for any positive integers M, N .

We can write a recursive definition of the gcd function:

$$gcd(N, M) = \begin{cases} 1 & \text{if } N = 1 \\ 1 & \text{if } M = 1 \\ N & \text{if } M = N \\ gcd(M, N) & \text{if } N < M \\ gcd(N - M, M) & \text{otherwise} \end{cases}$$

2. Write a prime number definition in Prolog. Define a relation $prime(X)$ which is true if and only if X is a prime number for any positive integer X .
3. Quick sort is a sorting algorithm that arranges elements in a list in ascending order. Given a list L , the algorithm applies divide-and-conquer:

- (a) Taking the first element A in L as the pivot
- (b) Partitions the remaining list L into two parts: P_1 and S_1 , such that P_1 contains all elements in L that are smaller than or equal to A , and S_1 contains all elements in L that are bigger than A .
- (c) Recursively sort P_1 and S_1 to produce two sorted lists P_2 and S_2 , respectively.
- (d) Then append the resulting lists P_2 with A and with S_2 , and return the result.

Write a Prolog implementation of quick sort. You may want to implement the following:

- $partition(A, List, S_1, P_1)$: this is the partition relation that split $List$ into two sublists S_1 and P_1 using A as the pivot.
- $quicksort(List, List')$: This is the quicksort relation that is true when $List'$ is the sorted copy of $List$.

You may also use the *append* relation defined in class.