```
def add(u, v, b):
    n = len(u)
    w = [0] * (n + 1)
    k = 0
    for j in range(n - 1, -1, -1):
        w[j + 1] = (u[j] + v[j] + k) % b
        k = (u[j] + v[j] + k) // b
    w[0] = k
    return w

def subtract(u, v, b):
    n = len(u)
    w = [0] * n
    k = 0
    for j in range(n - 1, -1, -1):
        w[j] = (u[j] - v[j] + k) % b
        k = (u[j] - v[j] + k) // b
    return w

def multiply(u, v, b):
    n = len(u)
    m = len(v)
    w = [0] * (m + n)
    for j in range(m - 1, -1, -1):
        if v[j] == 0:
            w[j] = 0
            continue
        k = 0
        for i in range(n - 1, -1, -1):
            t = u[i] * v[j] + w[i + j + 1] + k
            w[i + j + 1] = t % b
            k = t // b
        w[j] = k
    return w

def fast_column(u, v, b):
    n = len(u)
    m = len(v)
    w = [0] * (m + n)
    t = 0
    for s in range(m + n):
        for i in range(s + 1):
            if n - i - 1 >= 0 and m - s + i - 1 >= 0:
                t += u[n - i - 1] * v[m - s + i - 1]
        w[m + n - s - 1] = t % b
        t = t // b
    return w

def divide(u, v, b):
    n = len(u)
    t = len(v)
    q = [0] * (n - t + 1)
    while compare(u, shift_left(v, n - t, b)) >= 0:
        q[n - t] += 1
        u = subtract(u, shift_left(v, n - t, b), b)
    for i in range(n, t, -1):
        if u[i] >= v[t - 1]:
            q[i - t - 1] = b - 1
        else:
            q[i - t - 1] = (u[i] * b + u[i - 1]) // v[t - 1]
        while q[i - t - 1] * (v[t - 1] * b + v[t - 2]) > u[i] * b**2 + u[i - 1] * b + u[i - 2]:
            q[i - t - 1] -= 1
        u = subtract(u, multiply_by_digit(shift_left(v, i - t - 1, b), q[i - t - 1], b), b)
        if u[0] < 0:
            u = add(u, shift_left(v, i - t - 1, b), b)
            q[i - t - 1] -= 1
    r = u
    return q, r
```