

Oct 17, 04 12:12

polynomialring.cpp

Page 1/1

```

#include <list>
#include "polynomialring.h"

typedef std::list<Polynomial> PolynomialRing;

PolynomialRing* generateElements(int modulus, int max_order) {
    PolynomialRing *r = new PolynomialRing;
    //int nCoeffs = (int)pow(modulus, max_order+1);
    int *coeffs = new int[max_order+1];

    for (int i = 0; i <= max_order; i++) coeffs[i] = 0;

    int cur_order = 0; // This only increases.
    while (true) {
        Polynomial *p = new Polynomial(cur_order);
        p->modulus = modulus;
        for (int i = 0; i <= cur_order; i++) {
            p->coefficients[i] = coeffs[i];
        }
        r->push_back(*p);

        for (int i = 0; i <= max_order; i++) {
            coeffs[i]++;
            if (i > cur_order) cur_order = i;
            if (coeffs[i] == modulus && i == max_order-1) {
                delete coeffs;
                return r;
            } else if (coeffs[i] == modulus)
                coeffs[i] = 0;
            else
                break;
        }
    }
}

```