HOMEWORK 2
Due: 05.04.19

1 Fast polynomial gcd

Let a and b be polynomials in $K[x]$, $\deg(a) = n$ and $\deg(b) = n-1$. The goal of this exercise is to develop an algorithm that computes $\gcd(a,b)$ in time $\mathcal{O}(M(n)\log^2 n)$. Let $(r_i)_i \in K[x]$ be the sequence of remainders produced by Extended Euclidean Algorithm (EEA), and $(q_i)_i$ - the sequence of quotients, i.e.,

$$r_{i-1} = q_i r_i + r_{i+1}, \quad \text{with } r_0 = a, r_1 = b, r_N = \gcd(a,b).$$

We shall furthermore assume that $\deg(r_i) = \deg(r_{i-1}) - 1$ for all i, i.e., the degree of the remainders decreases strictly by 1. This is merely to simplify the arguments, the idea works in general.

1. Re-write the EEA algorithm as a sequence of 2×2 matrix-vector multiplications of the form

$$M_i \cdot \begin{bmatrix} r_{i-1} \\ r_i \end{bmatrix}.$$

   Give an explicit form of M_i's.

2. We will first design a divide-and-conquer algorithm that gives the last term in the remainder sequence whose degree is more than $\deg(a)/2$, i.e, $r_{\lceil \deg(a)/2 \rceil}$.

   The algorithm relies on the idea that the quotient of two polynomials of degrees d_1 resp. d_2 depends only on the leading $\min\{d_1 - d_2 + 1, d_2\}$ terms of the divisor and the leading $d_1 - d_2 + 1$ terms of the dividend. More formally, consider two polynomials

$$a(x) = a_1(x)x^k + a_2(x)$$
$$b(x) = b_1(x)x^k + b_2(x),$$

   where $\deg(a_2) < k$ and $\deg(b_2) < k$. Let

$$a(x) = q(x)b(x) + r(x)$$
$$a_1(x) = q_1(x)b_1(x) + r_1(x),$$

   where $\deg(r) < \deg(b)$ and $\deg(r_1) < \deg(b_1)$. Show that if $\deg(b_1) \geq \frac{1}{2}\deg(a_1)$, which implies that $k \leq 2\deg(b) - \deg(a) = n - 2$, then

   1. $q(x) = q_1(x)$
   2. $r(x)$ and $r_1(x)x^k$ agree in all terms of degree $k+1$ or higher.

3. Using the notation

$$M_{i,j}^{a,b} = \begin{cases} \mathbb{I}_2, & i = j; \\ \begin{bmatrix} 0 & 1 \\ 1 & -q_j \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & -q_{j-1} \end{bmatrix} \cdot \ldots \cdot \begin{bmatrix} 0 & 1 \\ 1 & -q_{i+1} \end{bmatrix} & i < j, \end{cases}$$

and the previous question, argue that

$$M_{0,\lceil (n+k)/2 \rceil}^{a,b} = M_{0,\lceil (n-k)/2 \rceil}^{a_1,b_1}.$$

4. Consider the following Hgcd ("Half-GCD") algorithm that takes two polynomials $a, b \in K[x]$ and returns a matrix $M_{0,\lceil n/2 \rceil}^{a,b}$ which yields the remainder $r_{\lceil n/2 \rceil}$.

```
1: function HGCD(a, b)
2:     if deg b ≤ deg a/2 then
3:         Return [ 1  0' ]
               [ 0  1  ]
4:     end if
5:     m = ⌈deg(a)/2⌉
6:     f ← a quo x^m, g ← b quo x^m
7:     M ←HGCD(f, g)
8:     [ a' ] ← M [ a ]
       [ b' ]     [ b ]
9:     c' ← a' mod b'
10:    M' ← [ 0          1          ]
            [ 1   −(a' quo b')     ]
11:    b'' ← b' quo x^m, c'' ← c' quo x^m
12:    M'' ←HGCD(b'', c'')
13:    Return M''M'M
14: end function
```

Using question 2, show its correctness. Argue that the complexity of this algorithm is $\mathcal{O}(M(n) \log^2 n)$.

5. Describe a recursive fast polynomial GCD algorithm of complexity $\mathcal{O}(M(n) \log^2 n)$ that uses HGCD() as a subroutine.

2 Primality Testing

Let n be an odd integer.

1. Show that n is prime if and only if $(X+1)^n \equiv X^n + 1 \bmod n$.

2. We assume that $n = p^a k$ with $p \nmid k$ and $k > 1$. Show that $p \nmid \binom{n}{p^a}$ and deduce that $(X+1)^n \not\equiv X^n + 1 \bmod p$

3. Let $\ell \geq 1$. Show that $P_p(X) = ((X+1)^n - X^n - 1) \bmod p$ has at most $\lfloor n/\ell \rfloor$ irreductible factors of degree ℓ in $\mathbb{Z}_p[X]$.

4. Explain why we cannot use Schwartz-Zippel Lemma to test the identity from question 1 and immediately get an efficient randomized algorithm for testing primality.

5. Let $\ell \geq 1$. We admit a result from Lidl and Niederreither (1986) that proves that for $n > p > 16$, the number I_ℓ of monic irreducible polynomials of degree ℓ in $\mathbb{Z}_p[X]$ is larger than $p^\ell/(2\ell)$.

   Show that for $n > p > 16$ and $\ell = \lceil \log_2 n \rceil$, the probability that a monic polynomial Q_p of degree ℓ uniformly drawn from $\mathbb{Z}_p[X]$ is irreducible but does not divide P_p is at least $1/(4\ell)$.

6. Show that the probability that a monic polynomial Q_n of degree ℓ uniformly drawn from $\mathbb{Z}_n[X]$ does not divide $P_n(X) = ((X+1)^n - X^n - 1) \bmod n$ is at least $1/(4\ell)$.

7. Conclude by proposing a randomized polynomial time algorithm that on input n outputs whether n is prime or not.