Notebook Swag

Sam Grayson

May 23, 2015

Throughout my homework, I will use $\sum^k a_i x^i$ to represent a polynomial of degree k with integer coefficients $a_1, a_2, \ldots, a_k$ where $a_k \neq 0$

Theorem: Let $f(x) = \sum^k a_i x^i$. $f(r) = 0 \leftrightarrow f(x) = (x - r) \cdot g(x)$ for some $g(x) = \sum^{k-1} b_i x^i$

Proof:

6.2 Theorem: Let $f(x) = \sum^k a_i x^i$. Let $f(r) \equiv 0$ for some $r \in \mathbb{Z}$. Let $g(x) = \sum^k b_i x^i$ where $b_i = a_i$ for $i > 0$ and $b_0 \equiv a_0$. Then there exists an $h(x) = \sum^{k-1} c_i x^i$ such that $(x-r)h(x) = g(x)$. (All congruences are taken mod n.)

Proof: $f(r) \equiv 0$. Let $b_0 := a_0 - f(r)$, since $b_0 \equiv a_0 - 0$ still holds. Then $g(r) = a_n r^n + \cdots + a_1 r + b_0 = a_n r^n + \cdots + a_1 r + a_0 - f(r) = f(r) - f(r) = 0$. Therefore we can apply 6.1 to $g(x)$. There exists an $h(x) = \sum^{k-1} c_i x^i$ where $(x - r)h(x) = g(x)$. ∎

Corollary: Let $f(x) = \sum^k a_i x^i$. If $f(r) \equiv 0$ for some $r \in \mathbb{Z}$, then there exists a $g(x) = \sum^k a_i x^i$ where $(x - r)g(x) \equiv f(x)$ for some $q \in \mathbb{Z}$. (All congruences are taken mod n.)

6.3 Theorem: $f(x) = \sum^n a_i x^i$, then $f(x) \equiv 0$ has at most n unique solutions modulo p. (All congruences are taken in a prime mod p.)

Proof: Assume there are no integer solutions to $f(x)$. Then the theorem is proven since $0 \leq n$.

Otherwise, $f(x)$ has at least one solution (call it r). Apply Corollary 6.2 to get $f(x) \equiv (x-r)g(x)$ where $g(x)$ is degree $k-1$. Assume $g(x)$ has no solution or its only solution is r. Then $x \neq r \rightarrow g(x) \neq 0$. But $x \neq r \rightarrow (x - r) \neq 0$, therefore $x \neq r \rightarrow f(x) \neq 0$. Therefore $f(x)$ has one solution. The theorem is proven since $1 \leq n$.

Otherwise, $g(x)$ has at least one solution (call it r). Apply Corollary 6.2 to get $g(x) \equiv (x - r)h(x)$ where $h(x)$ is degree $k - 2$. ...

Repeat until reaching "$n(x)$ has at least one solution (call it r)" where $n(x)$ is of degree 1. This takes n steps (each step reduces the degree by one starting at n), therefore there are at most n solutions to $f(x)$.

6.4 Theorem: For $a \in \mathbb{Z}$, $\gcd(i, \operatorname{ord}(a)) = 1 \rightarrow \operatorname{ord}(a^i) = \operatorname{ord}(a)$. (All congruences and orders are taken in a prime mod p)

Proof: $a^{\operatorname{ord}(a)} \equiv 1 \equiv 1^i \equiv (a^{\operatorname{ord}(a)})^i \equiv (a^i)^{\operatorname{ord}(a)}$. Therefore by Theorem 4.10 (only-if part), $\operatorname{ord}(a) | \operatorname{ord}(a^i)$. $\gcd(i, \operatorname{ord}(a))$.

6.5 Theorem: There are at most $\phi(d)$ solutions to $x^d \equiv 1$. (All congruences and orders are taken in a prime mod p).

Proof:

6.6 Theorem: Let $g \in \mathbb{Z} \ni \operatorname{ord}(g) = p-1$. $\{0, g, g^1, \ldots, g^{p-1}\} \in$ CRS. (All congruences, orders, and residue systems are taken in a prime modulo p.)

Proof: By Theorem 4.8, $\{g^1, g^2, \ldots, g^{p-1}\}$ are pairwise incongruent.. $1 < g < p \land p \in \mathbb{P}$, therefore $\gcd(g, p) = 1$. Then by Theorem 4.2, $\gcd(p, g^i) = 1$ for some $i \in \mathbb{Z}$. Therefore $g^i \not\equiv 0$. Therefore $\{0, g^1, g^2, \ldots, g^{p-1}\}$ are pairwise incongruent. By Theorem 3.16, $\{0, g, g^1, \ldots, g^{p-1}\} \in$ CRS. ■

6.7 Exercise: Find the primitive roots of the primes less than 20.

Code:

```
def order(a, n):
    # Calculate k where a^k ≡ 1 (mod n)
    if gcd(a, n) != 1:
        return None
    for k in count(1): # count up from one
        # if a^k ≡ 1 (mod n)
        if mod_exp(a, k, n, printing=False) == 1:
            # Using the modular exponentiation algorithm found in 3.6
            # k = ord_n(a)
            return k

def mod_exp(a1, r, n):
    # Returns the k in a^r ≡ k (mod n) where 0 ≤ k < r
    # This algorithm is found in 3.6
    # WLOG a < n
    a = cmod(a1, n) # reduce a mod n if possible
    a_squared = cmod(a * a, n)
    r_halved, remainder = division(r, 2)
    if r == 1:
        # Base case
        return a
    if divides(2, r):
        # (a^2)^(r/2)
        k = mod_exp(a_squared, r_halved, n)
        k = cmod(k, n) # reduce k mod n
        return k
    else:
        # (a^2)^((r-1)/2) · a
        k = mod_exp(a_squared, r_halved, n)
```

```
        ka = cmod(k * a, n)
        return ka

print(r'\begin{tabular}[t]{ll}')
print(r'\textbf{Mod} & \textbf{Primitive roots} \\')
for p in first(8, primes()):
    primitive_roots = []
    for a in range(1, p):
        if order(a, p) == p - 1:
            #print(r'{a} is a primitive root of {p} \\'.format(**locals()))
            primitive_roots.append(str(a))
    primitive_roots = ', '.join(primitive_roots)
    print(r'{a} & {primitive_roots} \\'.format(**locals()))
print(r'\end{tabular}')
```

Output:

Mod	**Primitive roots**
1	1
2	2
4	2, 3
6	3, 5
10	2, 6, 7, 8
12	2, 6, 7, 11
16	3, 5, 6, 7, 10, 11, 12, 14
18	2, 3, 10, 13, 14, 15

6.8 Theorem: Every prime has a primitive root.

Proof:

6.9 Exercise:

Code:

```
# using the 'order' function provided earlier

def powerset(iterable):
   # powerset([1,2,3]) --> () (1,) (2,) (3,) (1,2) (1,3) (2,3) (1,2,3)
    s = list(iterable)
    return chain.from_iterable(combinations(s, r) for r in range(len(s)+1))

def unique(iterable):
    # Returns all unique elements from the iterable
    seen = set()
    for element in iterable:
```

```
        if element not in seen:
            seen.add(element)
            yield element

def positive_factors(n):
    # Returns all positive numbers that divide n
    factors = []
    for primes_list in unique(powerset(prime_factorization(n))):
        factors.append(product(primes_list))
    return factors

print(r'\begin{tabular}[t]{ll} \\')
print(r'$d$ & \\'.format(**locals()))
for d in positive_factors(13 - 1):
    output = []
    for i in range(1, 13):
        if order(i, 13) == d:
            output.append(r'\circled{{{i}}}'.format(**locals()))
        else:
            output.append('{i}'.format(**locals()))
    output = ', '.join(output)
    print(r'{d} & $ \{{{output}\}} $ \\'.format(**locals()))
print(r'\end{tabular}')
```

Output:

d	
1	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
2	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
3	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
4	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
6	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
12	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

6.10 Exercise:

```
for n in [6, 10, 24, 36, 27]:
        # in python, the map function is like the image operator
        # map(f, set_a) asks for the image of f under the domain set_a
    s = sum(map(phi, positive_factors(n)))
        print(s)
        # here, I am maping phi over all positive factors of n and taking the sum
```

Output:

1. $\sum_{d|6} \phi(d) = \phi(1) + \phi(2) + \phi(3) + \phi(6)$
   $1 + 1 + 2 + 2 = 6$
2. $\sum_{d|10} \phi(d) = \phi(1) + \phi(2) + \phi(5) + \phi(10)$
   $1 + 1 + 4 + 4 = 10$
3. $\sum_{d|24} \phi(d) = \phi(1) + \phi(2) + \phi(3) + \phi(4) + \phi(6) + \phi(8) + \phi(12) + \phi(24)$
   $1 + 1 + 2 + 2 + 2 + 4 + 4 + 8 = 24$
4. $\sum_{d|36} \phi(d) = \phi(1) + \phi(2) + \phi(3) + \phi(4) + \phi(6) + \phi(9) + \phi(12) + \phi(18) + \phi(36)$
   $1 + 1 + 2 + 2 + 2 + 6 + 4 + 6 + 12 = 36$
5. $\sum_{d|27} \phi(d) = \phi(1) + \phi(3) + \phi(9) + \phi(27)$
   $1 + 2 + 6 + 18 = 27$

6.11 Lemma: $p \in \mathbb{P} \to \sum_{d|p} \phi(d) = p$.

Proof: The only divisors of a prime are 1 and p (see the definition of prime). $\phi(1) = 1$ (see note on the definition of ϕ) and $\phi(p) = p - 1$ as demonstrated earlier (Corollary 4.33). $\sum_{d|p} \phi(d) = \phi(1) + \phi(p) = 1 + p - 1 = p$.