POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG2810
STRUCTURES DISCRÈTES

TD 7 : THÉORIE DES NOMBRES

Directives pour la remise:

- La remise est individuelle et se fait à la fin de la séance de TD.
- Répondez directement sur ce document Word (docx).
- Lorsque vous avez terminé, générez un PDF avec le nom sous le format: ***SectionDeTD-Matricule.pdf*** (exemple: *04-1234567.pdf*).
- Téléchargez votre fichier PDF dans la boîte de remise située dans la Zone TDs de la page Moodle du cours.
- Choisissez la boîte de remise qui correspond à votre section de TD.
- Aucun retard et aucune remise par courriel ne seront accepté.
- Dans l'intérêt de l'équité pour tous les étudiants, vous devez modifier le fichier Word (docx) fourni. Modifier le fichier EXCLUT le fait d'intégrer des scans de rédaction manuscrite ou d'y écrire avec un stylet.
- Le non-respect des consignes entraînera automatiquement la note 0 pour ce TD.

Objectifs du TD7

Exercice 1: Calculer le plus grand commun diviseur de deux nombres à l'aide de la version possiblement la plus simple de l'algorithme d'Euclide (La version par soustraction).

Exercice 2: Calculer le plus grand commun diviseur ainsi que les entiers s et t du théorème de Bézout à l'aide de l'algorithme d'Euclide étendu.

Exercice 3: Mettre en pratique et appliquer le petit théorème de Fermat.

Exercice 4: Valider que la notion de congruence linéaire est bien comprise.

Exercice 5: Appliquer l'algorithme d'Euclide pour résoudre la congruence linéaire d'une manière très dirigée.

Exercice 6: Appliquer l'algorithme d'Euclide pour résoudre la congruence linéaire de manière autonome.

Exercice 7: Appliquer l'algorithme d'Euclide pour résoudre un système de congruence linéaire.

Exercice 8: Analyser et discuter les propriétés de l'algorithme d'Euclide.

La remise est individuelle mais le travail en équipe est encouragé. Veuillez inscrire votre nom, prénom et matricule ainsi que les noms des collègues avec lesquels vous avez collaboré pour le TD.

Nom:

Prénom:

Matricule:

Collègues:

Exercice 1

L'algorithme d'Euclide (vers 300 av. J.-C.) calcule le plus grand commun diviseur $pgcd(a, b)$ de deux nombres entiers a et b et est basé sur trois observations simples:

1. Pour $a = b$, le pgcd de a et b est a (ou b)
2. Pour $a > b$, d est un diviseur commun de a et b si et seulement si d est un diviseur commun de $a - b$ et b.
3. Pour $a < b$, d est un diviseur commun de a et b si et seulement si d est un diviseur commun de $b - a$ et a.

Algorithme 1.

```
pgcd(a, b: entiers positifs)
    if a == b:
        return a
    else if a > b:
        return pgcd(a - b, b)
    else:
        return pgcd(a, b - a)
```

L'algorithme utilise ainsi une succession de soustractions alternées et retourne le pgcd de a et b.

Utilisez l'algorithme 1 afin de calculer les $pgcd$ suivants en détaillant votre calcul.

a) $pgcd(5,\ 7)$

Réponse:

$pgcd(5,\ 7) = pgcd(5,\ 2) = pgcd(3,\ 2) = pgcd(1,\ 2) = pgcd(1,\ 1)=1$

b) $pgcd(8,\ 12)$

Réponse:

$pgcd(8,\ 12) = pgcd(8,\ 4) = pgcd(4,\ 4)=4$

Exercice 2

L'algorithme d'Euclide étendu $epgcd$, encore appelé théorème de Bézout, retourne également le nombre de fois qu'il a fallu additionner ou soustraire a et b pour obtenir le plus grand commun diviseur d.

$$sa + tb = d$$

Algorithme 2.

```
epgcd(A, B: vecteurs)
        if A[0] == B[0]:
                return A
        else if A[0] > B[0]:
                return epgcd(A - B, B)
        else:
                return epgcd(A, B - A)
```

L'astuce est de remplacer les nombres entiers a et b de l'algorithme 1 de l'exercice précédent par les vecteurs A = [a, 1, 0] et B = [b, 0, 1]. L'algorithme retourne le vecteur [d, s, t].

Notes:

1. Si A = [x, y, z] et B = [r, s, t] alors A - B = [x - r, y - s, z - t].
2. Les solutions alternatives [d, s + x, t + y] peuvent s'obtenir ensuite avec les solutions de $xa + yb = 0$, par exemple x = b et y = -a.

Utilisez l'algorithme 2 afin d'obtenir [d, s, t] pour les nombres suivants et vérifiez que le résultat satisfait l'équation $sa + tb = d$.

a) epgcd([9, 1, 0], [6, 0, 1])

Réponse:

1. [9, 1, 0], [6, 0, 1]
2. [3, 1, -1], [6, 0, 1]
3. [3, 1, -1], [3, -1, 2]

Vérification: 1 x 9 + (- 1) x 6 = 3

b) epgcd([3, 1, 0], [5, 0, 1])

Réponse:

1. [3, 1, 0], [5, 0, 1]
2. [3, 1, 0], [2, -1, 1]
3. [1, 2, -1], [2, -1, 1]
4. [1, 2, -1], [1, -3, 2]

Vérification: 2 x 3 + (-1) x 5 = 1

Exercice 3

Utilisez le petit théorème de Fermat pour calculer

a) $3^{302} \mod 5$

Réponse:
Par le petit théorème de Fermat nous savons que $3^4 \equiv 1 \pmod 5$, alors $3^{300} = (3^4)^{75} \equiv 1 \pmod 5$ et donc $3^{302} = 3^2 \cdot 3^{300} \equiv 9 \cdot 1 = 9 \pmod 5$ que l'on peut réduire à $4 \pmod 5$.

b) $3^{302} \mod 7$

Réponse:
De manière similaire, $3^6 \equiv 1 \pmod 7$, alors $3^{300} = (3^6)^{50} \equiv 1 \pmod 7$ et donc $3^{302} = 3^2 \cdot 3^{300} \equiv 9 \cdot 1 = 9 \pmod 7$ que l'on peut réduire à $2 \pmod 7$.

Exercice 4.

Une congruence linéaire peut s'écrire sous la forme suivante:

$$a\,x \equiv b \pmod{m}$$

où a, b sont deux nombres entiers, m est un entier positif et x est une valeur entière inconnue que vous devez trouver.

Soit la congruence linéaire:

$$6x \equiv 4 \pmod{10}$$

Trouvez par essais et erreurs (en essayant 0, 1, 2, 3, …, 9) les solutions de la congruence linéaire (sachant que x mod 10 retourne tout simplement l'unité de x).

Réponse:

Tout entier de la forme $4 + 10k$ et $9 + 10k$ sont des solutions. Ou encore mieux nous pouvons écrire $x = 4 + 5k$.

Exercice 5

Résoudre la congruence linéaire $a\,x \equiv b \pmod{m}$ revient à résoudre:

$$a\,x + m\,y = b$$

où y peut prendre toutes les valeurs entières possibles.

Puisque nous savons déjà résoudre un tel système avec l'algorithme d'Euclide nous pouvons tout simplement appliquer ce que nous savons.

Théorème 1.

Soit $d = pgcd(a, m)$. Si d ne divise pas b alors la congruence linéaire $a \equiv b \pmod{m}$ n'a aucune solution. Si d divise b alors la congruence linéaire possède exactement d solutions, où par solution nous entendons une classe de congruence $\bmod m$.

De plus, toutes les solutions forment une seule classe de congruence $\bmod \frac{m}{d}$.

Utilisez maintenant l'algorithme d'Euclide étendu afin de résoudre la même congruence linéaire que le problème précédent.

$$6x \equiv 4 \pmod{10}$$

a) Résoudre cette congruence revient à résoudre $6x + 10y = 4$. En divisant par $d = pgcd(6,10) = 2$ l'équation devient $3x + 5y = 2$. À l'exercice 2b, vous avez trouvé une solution pour $3s + 5t = 1$. Pour résoudre l'équation $3x + 5y = 2$, il suffit ensuite de multiplier chacun des

termes par 2. Vérifiez que $x \equiv 2s \pmod 5$ sont bien les solutions du problème.

Réponse:

À l'exercice 2, nous avons trouvé que 3 x 2 + 5 x (-1) = 1 et donc $s = 2$. En utilisant ce résultat nous trouvons: $x = 2s = 4$

b) Une autre façon de résoudre $3x + 5y = 2$ que l'on peut réécrire $3x \equiv 2 \pmod 5$ est de trouver un inverse s tel que $3s \equiv 1 \mod 5$. Utilisez à nouveau la valeur de s trouvé à l'exercice 2b. Multipliez les deux côté de la congruence $3x \equiv 2 \pmod 5$ par s afin d'isoler x et obtenir la solution.

Réponse:

À l'exercice 2, nous avons trouvé que 3 x 2 + 5 x (-1) = 1 et donc $s = 2$. En utilisant ce résultat comme inverse multiplicatif nous trouvons: $x = 4$

Exercice 6

En utilisant les techniques travaillées à l'exercice précédent, trouvez l'ensemble des solutions possibles pour :

$$6x \equiv 9 \pmod{21}$$

Détaillez vos calculs.

Réponse:

Résoudre la congruence linéaire $6x \equiv 9 \pmod{21}$ revient à résoudre $6x + 21y = 9$. Le pgcd(6, 21) = 3. En divisant par 3 tous les termes de l'équation nous obtenons $2x + 7y = 3$.

En utilisant l'algorithme 2 nous trouvons d = 1, s = 4 et t = -1 que nous pouvons vérifier en substituant s et t dans l'équation: $2s + 7t = 1$.

Maintenant le terme de droite que nous recherchons n'est pas 1 mais bien 3, ainsi en multipliant s par 3 nous obtenons $x = 3s = 12$.

Nous pouvons donc écrire que les solutions sont: 12 + 7k où k est un nombre entier quelconque.

Il est aussi possible de résoudre le problème en utilisant l'inverse multiplicatif de 2 mod 7 qui est tout simplement $s = 4$. En multipliant les termes par s nous obtenons: $x \equiv 12 \pmod{7}$.

Exercice 7

Voici un exemple de la méthode des substitutions successives utilisée pour résoudre un système de congruences linéaires.

Soit le système:

$$x \equiv 8 \pmod{12}$$
$$x \equiv 6 \pmod{13}$$

Les solutions possibles pour la première congruence sont:

$$x = 8 + 12k_1$$

Si on substitue dans la deuxième congruence nous obtenons:

$$8 + 12k_1 \equiv 6 \pmod{13}$$
$$12k_1 \equiv -2 \pmod{13}$$
$$12k_1 \equiv 11 \pmod{13}$$

Le pgcd(12, 13) = 1 et 1 divise 11, donc selon le théorème 1 de l'exercice 5, le système a bien une solution. En utilisant l'algorithme d'Euclide étendu nous trouvons que s = -1 et t = 1. Nous pouvons donc utiliser -1 comme inverse de 12 mod 13 et ainsi trouver:

$$k_1 \equiv -11 \pmod{13}$$
$$k_1 \equiv 2 \pmod{13}$$

Alors $k_1 = 2 + 13k_2$ et en substituant dans l'équation pour x:

$$x = 8 + 12(2 + 13k_2)$$

$$x = 32 + 156k_2$$

Et donc les solutions du sytème sont: $x \equiv 32 \pmod{156}$.

Utilisez la méthode des substitutions successives, en suivant les étapes de l'exemple précédent, pour résoudre le système:

$$x \equiv 4 \pmod{7}$$
$$x \equiv 2 \pmod{11}$$

Réponse:

Les solutions possibles pour la première congruence sont:

$$x = 4 + 7k_1$$

Si on substitue dans la deuxième congruence nous obtenons:

$$4 + 7k_1 \equiv 2 \pmod{11}$$
$$7k_1 \equiv -2 \pmod{11}$$
$$7k_1 \equiv 9 \pmod{11}$$

Le pgcd(7, 11) = 1 et 1 divise 9, donc le système a une solution. En utilisant l'algorithme d'Euclide étendu nous trouvons que s = -3 et t = 2. Nous pouvons utiliser -3 comme inverse de 7 mod 11 et ainsi trouver:

$$k_1 \equiv -27 \pmod{11}$$

$$k_1 \equiv 6 \pmod{11}$$

Alors $k_1 = 6 + 11k_2$ et en substituant dans l'équation pour x:

$$x = 4 + 7(6 + 11k_2)$$
$$x = 46 + 77k_2$$

Et donc les solutions du sytème sont: $x \equiv 46 \pmod{77}$.

Exercice 8

a) Présentez un argument afin de démontrer que l'algorithme 1 se termine pour tout entier positif a et b.

Réponse:

Une suite strictement décroissante d'entiers positifs $a_0 > a_1 > a_2 > \cdots$ doit être finie. Puisque la procédure itérative qui vient d'être décrite produit une séquence strictement décroissante, les itérations doivent finalement s'arrêter, ce qui signifie qu'à un certain point a et b seront égaux, et que cette valeur est donc le pgcd de a et b. Le pire scénario est d'arrêter sur la valeur 1 mais l'algorithme peut s'arrêter avant.

b) Proposez une amélioration à l'algorithme 1 afin de diminuer le nombre d'itérations nécessaire pour obtenir le résultat. (Par exemple pour trouver le plus grand commun diviseur de 2 et 1000).

Réponse:

Remplacer les opérations de soustraction qui retourne la différence de deux nombres par l'opération modulo qui retourne le reste de la division de deux nombres.

```
pgcd(a, b: entiers positifs)
    if a == b:
        return a
    if a > b:
        pgcd(a  mod b, b)
    else:
        pgcd(a, b  mod a)
```