Préparation à l'agrégation de mathématiques
TP Sage Gauss-Bareiss et méthodes modulaires

1) Gauss classique

a) Ecrire une procédure gauss(M) qui renvoie la matrice obtenue par application du pivot de Gauss classique pour une matrice inversible (sans phase de « remontée »).
On pourra utiliser les commandes sage suivantes :
A=copy(M) [pour éviter que Sage n'effectue les manipulations sur la matrice d'origine]
A.nrows()
A.swap_rows(i,j)
A.add_multiple_of_row(i,j,alpha)

b) Tester sur une matrice générée aléatoirement, à coefficients dans Q
On pourra utiliser la commande random_matrix(QQ,n)

c) Ecrire une procédure qui renvoie le max des valeurs absolues des numérateurs et des dénumérateurs d'une matrice ; cette fonction mesure donc la taille des données de la matrices
Tester sur les matrices obtenues après application de l'algorithme de Gauss

d) On veut étudier comment augmente ces tailles avec la dimension n des matrices. On va faire varier cette dimension en prenant $n=2^k$ pour des valeurs de k consécutives croissantes (jusqu'à atteindre les limites de calcul de Sage, en un temps raisonnable). On travaille donc en base 2, et ce choix de valeurs de n permet de balayer une grande amplitude de nombres. Pour chaque application de l'algorithme de Gauss, on va calculer le nombre de chiffres de la taille des données de la matrice obtenue, en base 2 : il s'obtient par la commande x.ndigits(2).

Représenter le nuage de points obtenu en représentant ce nombre de chiffres en fonction de l'entier k. On utilisera la commande Sage points(enumerate(liste))

Représenter ensuite le nuage de points obtenu en représentant le logarithme en base 2 de ce nombre de chiffres, toujours en fonction de k. Qu'observez-vous ? Est-ce cohérent avec l'hypothèse d'un nombre de chiffres des numérateurs et dénominateurs polynomial en la dimension de la matrice (annoncé dans le cours) ?

e) On veut maintenant observer si les temps de calcul reflètent la complexité binaire annoncée dans le cours. Pour cela, on dispose de différentes commandes Sage :
- Essayer timeit('gauss(M)')
- On dispose de commandes supplémentaires après avoir chargé la librairie time (via la commande import time)

Programme permettant d'obtenir le temps de calcul du temps :

```
def temps_gauss(n):
    M=matrice_inversible(n) % cette procédure renvoie une matrice inversible de taille n générée aléatoirement
    debut=time.time()
    gauss(M)
    return(time.time() - debut)
```

Comme précédemment, faire varier la taille n sous la forme $n=2^k$ et tracer un graphique permettant de tester l'hypothèse d'une complexité polynomiale en la dimension de la matrices

2) Gauss entier

a) écrire une procédure gauss_ent(M) qui renvoie cette fois la matrice obtenue en appliquant l'algorithme de Gauss sans effectuer de divisions (algorithme présenté en cours).
Ceci nécessite de modifier un peu la procédure gauss déjà écrite. On pourra utiliser la commande Sage A.rescale_row(i,alpha)

Tester avec la matrice obtenue par la commande suivante :
M=random_matrix(ZZ,5,x=10,y=99)

Observer l'augmentation de la taille des coefficients selon l'étape de l'algorithme : est-ce conforme à ce qui a été vu en cours ?

3) Gauss-Bareiss
a) Il s'agit maintenant d'adapter le programme précédent pour obtenir une implémentation de l'algorithme de Gauss-Bareiss.
Ecrire la procédure gauss_b(M) correspondante
Attention : Sage n'accepte pas la division (même si c'est une division entière) ; pour y remédier, on pourra changer l'anneau dans lequel on travaille (passer de Z à Q) via la commande Sage suivante :
A=copy(M).change_ring(QQ) [à la place de A=copy(M)]

b) Calculer le déterminant de la matrice par une commande sage. Que remarquez-vous ? (observez bien la matrice obtenue en sortie de Gauss-Bareiss). Tester votre conjecture avec différentes matrices. En déduire une procédure gauss_b_det(M), basée sur Gauss-Bareiss, qui calcule le déterminant d'une matrice à coefficients dans Z (inversible dans Mn(Q))

4) Méthodes modulaires
a) Ecrire une procédure gauss_det(M) qui calcule le déterminant d'une matrice (à coefficients dans un anneau A, supposé inversible dans Mn(K) où K est le corps des fractions de A) en se basant sur le pivot de Gauss classique (comment est modifié le déterminant pour chaque type d'opération élémentaire sur les lignes?)

Tester la procédure avec une matrice M=random_matrix(ZZ,5,x=10,y=99) et comparer avec le résultat de gauss_b_det(M)

Tester également avec des matrices à coefficients dans un corps fini (générées aléatoirement avec la commande indiquée plus haut).

b) Méthode modulaire à 1 grand nombre premier
Ecrire une pocédure det_mod1(M) qui renvoie le déterminant calculé selon la méthode modulaire à un seul nombre premier. On utilisera la majoration a priori du résultat vu en cours (application de la formule de Hadamard).
Commandes Sage utiles : P=Primes(); p=P.next(N); [donne le premier nombre premier plus grand que N]
On peut passer d'un domaine de nombres à un autre par des commandes du type :
d=ZZ(gauss_det(A.change_ring(GF(p))))

Tester votre programme

c) Méthode à plusieurs nombres premiers
Ecrire maintenant une procédure det_mod(M) qui calcule le déterminant selon la méthode modulaire à plusieurs petits nombres premiers.
Commandes Sage utiles : P=Primes(); L=[P.unrank(i) for i in range(N)]
Pour mémoire, la commande crt résoud le problème des restes chinois

Tester votre programmes

d) On souhaite maintenant effectuer des comparaisons entre les 4 programmes : Gauss classique, Gauss-Bareiss, la méthode modulaire à 1 (resp. plusieurs) nombres premiers, pour calculer le déterminant.
A chaque fois, on pourra jouer sur les deux paramètres : la dimension n de la matrice et la taille des coefficients de la matrice.

Comparer avec M=random_matrix(ZZ,5,x=10,y=99);

Comparer avec M=random_matrix(ZZ,5,x=10^1000,y=10^1001)

Comparer avec M=random_matrix(ZZ,20,x=10^5,y=10^6)

Effectuer d'autres comparaisons

Est-ce que les différences au niveau des temps de calcul reflètent les différences au niveau de la complexité binaire (calculée en cours pour Gauss-Bareiss et indiquée dans le cours pour les méthodes modulaires) ?