

Dominos

On commence comme d'habitude par calculer un certain nombre de termes de la suite (u_n) :

```
u = [13]
for i in range(10000):
    u.append((15091 * u[-1]) % 64007)
```

Question 1.

```
def domino(k, M):
    x, y = u[2 * k] % (M + 1), u[2 * k + 1] % (M + 1)
    return (min(x, y), max(x, y))
```

Question 2.

```
def d(M, N):
    dominos = []
    k = 0
    while len(dominos) < N:
        d = domino(k, M)
        if d not in dominos:
            dominos.append(d)
        k += 1
    return k, dominos[-1]
```

Pour la suite du sujet, on redéfinit cette fonction pour renvoyer l'ensemble $D_{M,N}$ en remplaçant la dernière ligne par :

```
return dominos
```

Question 3.

```
def longueur2(dominos):
    s = 0
    for i in range(len(dominos)):
        for j in range(i + 1, len(dominos)):
            if dominos[i][0] in dominos[j] or dominos[i][1] in dominos[j]:
                s += 1
    return s
```

Question 4. Notons (g, d) le domino de rang n . Alors $L_{n,v} = \begin{cases} \max(1 + L_{n-1,d}, L_{n-1,v}) & \text{si } v = g \\ \max(1 + L_{n-1,g}, L_{n-1,v}) & \text{si } v = d \\ L_{n-1,v} & \text{sinon} \end{cases}$

Question 5.

```
def plusLongTrainOrdonne(M, N):
    dominos = d(M, N)
    L = [[0 for _ in range(M + 1)] for _ in range(N + 1)]
    for n in range(1, N + 1):
        (x, y) = dominos[n - 1]
        for v in range(M + 1):
            if x == v:
                L[n][v] = max(1 + L[n - 1][y], L[n - 1][v])
            elif y == v:
                L[n][v] = max(1 + L[n - 1][x], L[n - 1][v])
            else:
```

```
    L[n][v] = L[n - 1][v]
return max(L[N])
```

Question 6. On commence par une fonction qui calcule le plus long train se terminant par v :

```
def aux(v, dominos):
    trainmax = 0
    for d in dominos:
        (x, y) = d
        if x == v:
            trainmax = max(trainmax, 1 + aux(y, [dd for dd in dominos if dd != d]))
        elif y == v:
            trainmax = max(trainmax, 1 + aux(x, [dd for dd in dominos if dd != d]))
    return trainmax
```

puis la fonction principale :

```
def plusLongTrain(M, N):
    dominos = d(M, N)[1]
    trainmax = 0
    for v in range(M + 1):
        trainmax = max(trainmax, aux(v, dominos))
    return trainmax
```