Nom : _____ (_____ /5)

PCP – Q07

Un collègue a développé une classe `Matrice<M,N,T>` où `M` est un entier, `N` est un entier et `T` est le type des éléments (typiquement : `double`). Une instance de `Matrice<M,N,T>` contient un attribut de type `array<array<T,N>,M>`, ce qui équivaut structurellement, pour l'essentiel du moins, à un tableau `T tab[M][N]`.

Ce collègue jongle avec deux algorithmes de multiplication matricielle, tous deux de complexité égale soit $O(n^3)$:

Algorithme A

```
template <int M, int N, int K, class T>
   Matrice<M,N,T> operator*(
      const Matrice<M,K,T> &m0,
      const Matrice<K,N,T> &m1
   )
{
   Matrice<M,N,T> res;
   for (int i = 0; i < M; ++i)
      for (int j = 0; j < N; ++j)
         for (int k = 0; k < K; ++k)
            res[i][j] += m0[i][k] * m1[k][j];
   return res;
}
```

Algorithme B

```
template <int M, int N, int K, class T>
   Matrice<M,N,T> operator*(
      const Matrice<M,K,T> &m0,
      const Matrice<K,N,T> &m1
   )
{
   Matrice<M,N,T> res;
   for (int i = 0; i < M; ++i)
      for (int k = 0; k < K; ++k)
         for (int j = 0; j < N; ++j)
            res[i][j] += m0[i][k] * m1[k][j];
   return res;
}
```

Ce qui rend votre collègue perplexe est ceci : il a effectué divers tests des deux algorithmes et il se trouve que l'algorithme `A` est *beaucoup* plus lent que l'algorithme `B` (avec 1000 tests sur des matrices 100×100, par exemple, l'algorithme `A` prend plus de 38 secondes à s'exécuter alors qu'il n'en faut qu'environ 1,1 pour exécuter l'algorithme `B`).

Expliquez, de la manière la plus claire possible, la raison de cet écart.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____