

Exercício: Multiplicação de matrizes usando Barreira Cíclica + Thread Pools.

Este trabalho tem que ser feito, necessariamente, usando thread pools e barreira cíclica.

Também deve ser medido o tempo de cálculo da multiplicação de matrizes. Você pode se basear em slides anteriores que tinha esse tipo de cálculo (**System.currentTimeMillis()**).

Defina um valor de N constante, e gere duas matrizes inteiras aleatórias A e B, ambas de dimensões N x N, com números entre 0 e 100.

Defina um **thread pool de NxN threads**, onde cada thread deve gerar o valor da multiplicação de **uma linha** da matriz A por **uma coluna** da matriz B e armazenar esse valor na matriz C (resultado da multiplicação).

Defina uma barreira cíclica também de **NxN threads**. Dê **um nome** para cada uma dessas threads (pode ser Thread i j, onde i é a linha e j é a coluna). Além disso, as threads precisam chamar o **await()** da barreira cíclica indicando que a thread chegou na barreira. A **barreira cíclica** pode ser um atributo da thread e deve ser passado como parâmetro ao construtor da thread.

Note que, como explicado em sala de aula, o **construtor** da barreira cíclica, além da quantidade de threads (NxN), também inclui um comando. Este comando pode ser a **instanciação de uma thread X**. No caso, essa thread X será automaticamente chamada quando todas as NxN threads chegarem na barreira. O objetivo dessa thread X será simplesmente **imprimir as matrizes A, B, C e o tempo** de cálculo da multiplicação de matrizes.

Implemente tudo em um único arquivo Java.