

Intelligent Coding

Memory Allocation and Deallocation: You correctly allocate memory for matrices A, B, and C using dynamic allocation (`new`). But don't forget to deallocate this memory using `delete[]` after you finish using these matrices to avoid memory leaks.

Random Number Generation: Your current method for generating random numbers using `rand()` and `srand(time(0))` works fine, but it's not the most modern or efficient approach. Consider using the `<random>` library for better random number generation.

Output Precision: When printing the time taken for matrix multiplication, you might want to set the precision of the output stream using `std::fixed` and `std::setprecision` for better readability.

Loop Interchange: In your code, you're swapping the loops for matrix multiplication to analyze different loop orders. This is good for understanding performance differences, but in practical scenarios, compilers are often smart enough to optimize loop orders on their own. So, the impact of manual loop reordering might be different in real-world scenarios.

Benchmarking Environment: Ensure that you're running your benchmarks in a controlled environment, preferably without other heavy processes running in the background, to get accurate timing results.

Optimization Flags: Compile your code with optimization flags (`-O2`, `-O3`) to allow the compiler to perform various optimizations, which might affect the performance results.

```
sachin12345@sachin12345-VirtualBox: ~  
g++: error: 03: No such file or directory  
sachin12345@sachin12345-VirtualBox:~$ g++ -O3 -o m m.cpp  
sachin12345@sachin12345-VirtualBox:~$ ./m  
Time taken for matrix multiplication with ijk loop order: 4.78358 seconds  
Time taken for matrix multiplication with jik loop order: 4.47623 seconds  
Time taken for matrix multiplication with kij loop order: 1.55117 seconds  
sachin12345@sachin12345-VirtualBox:~$ ./m  
Time taken for matrix multiplication with ijk loop order: 4.7008 seconds  
Time taken for matrix multiplication with jik loop order: 4.1089 seconds  
Time taken for matrix multiplication with kij loop order: 1.31941 seconds  
sachin12345@sachin12345-VirtualBox:~$ ./m  
Time taken for matrix multiplication with ijk loop order: 4.75569 seconds  
Time taken for matrix multiplication with jik loop order: 3.95875 seconds  
Time taken for matrix multiplication with kij loop order: 1.30417 seconds  
sachin12345@sachin12345-VirtualBox:~$ ./m  
Time taken for matrix multiplication with ijk loop order: 4.67123 seconds  
Time taken for matrix multiplication with jik loop order: 4.03484 seconds  
Time taken for matrix multiplication with kij loop order: 1.33703 seconds  
sachin12345@sachin12345-VirtualBox:~$ ./m  
Time taken for matrix multiplication with ijk loop order: 4.76058 seconds  
Time taken for matrix multiplication with jik loop order: 4.17154 seconds  
Time taken for matrix multiplication with kij loop order: 1.32929 seconds  
sachin12345@sachin12345-VirtualBox:~$ ./m  
Time taken for matrix multiplication with ijk loop order: 4.86437 seconds  
Time taken for matrix multiplication with jik loop order: 4.16674 seconds  
Time taken for matrix multiplication with kij loop order: 1.28009 seconds  
sachin12345@sachin12345-VirtualBox:~$ ./m  
Time taken for matrix multiplication with ijk loop order: 4.92792 seconds  
Time taken for matrix multiplication with jik loop order: 4.00119 seconds  
Time taken for matrix multiplication with kij loop order: 1.31548 seconds  
sachin12345@sachin12345-VirtualBox:~$
```

Figure 1

