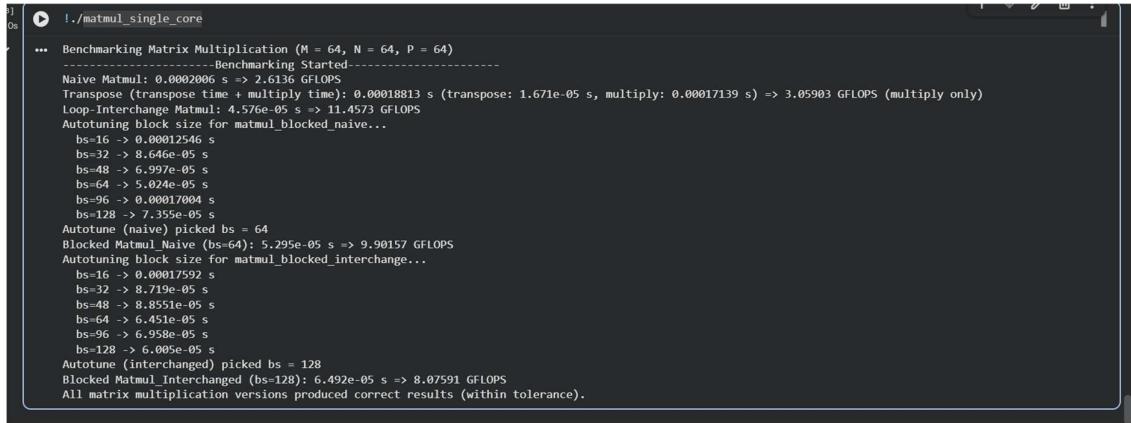# Benchmarking – Different Cases
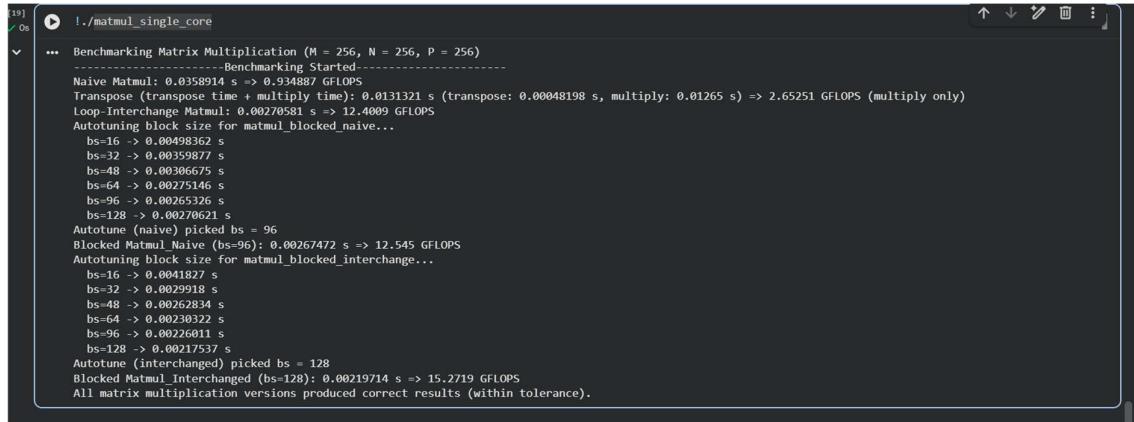
1. **Tiny / L1-friendly** — 64 × 64 × 64
   Purpose: sanity check, correctness, fastest small-run behavior (fits in L1).


```
!./matmul_single_core

Benchmarking Matrix Multiplication (M = 64, N = 64, P = 64)
---------------------Benchmarking Started---------------------
Naive Matmul: 0.0002006 s => 2.6136 GFLOPS
Transpose (transpose time + multiply time): 0.00018813 s (transpose: 1.671e-05 s, multiply: 0.00017139 s) => 3.05903 GFLOPS (multiply only)
Loop-Interchange Matmul: 4.576e-05 s => 11.4573 GFLOPS
Autotuning block size for matmul_blocked_naive...
    bs=16 -> 0.00012546 s
    bs=32 -> 8.646e-05 s
    bs=48 -> 6.997e-05 s
    bs=64 -> 5.024e-05 s
    bs=96 -> 0.00017004 s
    bs=128 -> 7.355e-05 s
Autotune (naive) picked bs = 64
Blocked Matmul_Naive (bs=64): 5.295e-05 s => 9.90157 GFLOPS
Autotuning block size for matmul_blocked_interchange...
    bs=16 -> 0.00017592 s
    bs=32 -> 8.719e-05 s
    bs=48 -> 8.8551e-05 s
    bs=64 -> 6.451e-05 s
    bs=96 -> 6.958e-05 s
    bs=128 -> 6.005e-05 s
Autotune (interchanged) picked bs = 128
Blocked Matmul_Interchanged (bs=128): 6.492e-05 s => 8.07591 GFLOPS
All matrix multiplication versions produced correct results (within tolerance).
```

2. **Small / L2-friendly** — 256 × 256 × 256
   Purpose: checks cache-working-set beyond L1 (tests loop ordering benefits).


```
!./matmul_single_core

Benchmarking Matrix Multiplication (M = 256, N = 256, P = 256)
---------------------Benchmarking Started---------------------
Naive Matmul: 0.0358914 s => 0.934887 GFLOPS
Transpose (transpose time + multiply time): 0.0131321 s (transpose: 0.00048198 s, multiply: 0.01265 s) => 2.65251 GFLOPS (multiply only)
Loop-Interchange Matmul: 0.00270581 s => 12.4009 GFLOPS
Autotuning block size for matmul_blocked_naive...
    bs=16 -> 0.00498362 s
    bs=32 -> 0.00359877 s
    bs=48 -> 0.00306675 s
    bs=64 -> 0.00275146 s
    bs=96 -> 0.00265326 s
    bs=128 -> 0.00270621 s
Autotune (naive) picked bs = 96
Blocked Matmul_Naive (bs=96): 0.00267472 s => 12.545 GFLOPS
Autotuning block size for matmul_blocked_interchange...
    bs=16 -> 0.0041827 s
    bs=32 -> 0.0029918 s
    bs=48 -> 0.00262834 s
    bs=64 -> 0.00230322 s
    bs=96 -> 0.00226011 s
    bs=128 -> 0.00217537 s
Autotune (interchanged) picked bs = 128
Blocked Matmul_Interchanged (bs=128): 0.00219714 s => 15.2719 GFLOPS
All matrix multiplication versions produced correct results (within tolerance).
```

3. **Medium / L3-friendly** — 512 × 512 × 512
   Purpose: typical in-memory compute case; shows blocked gains.

```
!./matmul_single_core

Benchmarking Matrix Multiplication (M = 512, N = 512, P = 512)
---------------------Benchmarking Started---------------------
Naive Matmul: 0.483542 s => 0.555144 GFLOPS
Transpose (transpose time + multiply time): 0.117796 s (transpose: 0.00250228 s, multiply: 0.115293 s) => 2.32829 GFLOPS (multiply only)
Loop-Interchange Matmul: 0.021956 s => 12.2261 GFLOPS
Autotuning block size for matmul_blocked_naive...
    bs=16 -> 0.0446957 s
    bs=32 -> 0.0317575 s
    bs=48 -> 0.0257917 s
    bs=64 -> 0.02313 s
    bs=96 -> 0.0243001 s
    bs=128 -> 0.0226375 s
Autotune (naive) picked bs = 128
Blocked Matmul_Naive (bs=128): 0.022495 s => 11.9331 GFLOPS
Autotuning block size for matmul_blocked_interchange...
    bs=16 -> 0.0400307 s
    bs=32 -> 0.0280462 s
    bs=48 -> 0.0226313 s
    bs=64 -> 0.0201238 s
    bs=96 -> 0.0194458 s
    bs=128 -> 0.0187793 s
Autotune (interchanged) picked bs = 128
Blocked Matmul_Interchanged (bs=128): 0.0187048 s => 14.3511 GFLOPS
All matrix multiplication versions produced correct results (within tolerance).
```

4. **Large / Memory-pressure** — 1024 × 1024 × 1024
   Purpose: approach L3 / main-memory limits on many machines — shows memory-bound behavior.

```
!./matmul_single_core

Benchmarking Matrix Multiplication (M = 1024, N = 1024, P = 1024)
---------------------Benchmarking Started---------------------
Naive Matmul: 5.85199 s => 0.366966 GFLOPS
Transpose (transpose time + multiply time): 0.990216 s (transpose: 0.00977808 s, multiply: 0.980438 s) => 2.19033 GFLOPS (multiply only)
Loop-Interchange Matmul: 0.167389 s => 12.8293 GFLOPS
Autotuning block size for matmul_blocked_naive...
    bs=16 -> 0.36766 s
    bs=32 -> 0.259493 s
    bs=48 -> 0.211529 s
    bs=64 -> 0.200189 s
    bs=96 -> 0.19303 s
    bs=128 -> 0.208639 s
Autotune (naive) picked bs = 96
Blocked Matmul_Naive (bs=96): 0.196648 s => 10.9204 GFLOPS
Autotuning block size for matmul_blocked_interchange...
    bs=16 -> 0.30975 s
    bs=32 -> 0.234979 s
    bs=48 -> 0.193482 s
    bs=64 -> 0.174011 s
    bs=96 -> 0.161539 s
    bs=128 -> 0.165242 s
Autotune (interchanged) picked bs = 96
Blocked Matmul_Interchanged (bs=96): 0.157814 s => 13.6077 GFLOPS
All matrix multiplication versions produced correct results (within tolerance).
```

5. **Very large / stress test** — 2048 × 2048 × 2048
   Purpose: heavy compute & memory; use for final best-known-tweak timings (might take long).

```
[35]   ▶   !./matmul_single_core
✓ 1m

   ⋯   Benchmarking Matrix Multiplication (M = 2048, N = 2048, P = 2048)
       ----------------------Benchmarking Started----------------------
       Naive Matmul: 67.1084 s => 0.256002 GFLOPS
       Transpose (transpose time + multiply time): 8.23425 s (transpose: 0.0456095 s, multiply: 8.18864 s) => 2.09801 GFLOPS (multiply only)
       Loop-Interchange Matmul: 4.05504 s => 4.23667 GFLOPS
       Autotuning block size for matmul_blocked_naive...
          bs=16 -> 4.21944 s
          bs=32 -> 2.68331 s
          bs=48 -> 1.95446 s
          bs=64 -> 1.82055 s
          bs=96 -> 1.89704 s
          bs=128 -> 1.94693 s
       Autotune (naive) picked bs = 64
       Blocked Matmul_Naive (bs=64): 1.83544 s => 9.36007 GFLOPS
       Autotuning block size for matmul_blocked_interchange...
          bs=16 -> 4.19837 s
          bs=32 -> 2.51499 s
          bs=48 -> 2.4046 s
          bs=64 -> 1.83855 s
          bs=96 -> 1.69815 s
          bs=128 -> 1.73419 s
       Autotune (interchanged) picked bs = 96
       Blocked Matmul_Interchanged (bs=96): 1.69947 s => 10.109 GFLOPS
       All matrix multiplication versions produced correct results (within tolerance).
```

6. **Rectangular / real-world pattern** — 2048 × 512 × 4096 (A: 2048×512, B: 512×4096 → C: 2048×4096)
   Purpose: tests non-square workloads (skinny K, wide N) — exposes different memory/compute balance.

```
[40]   ▶   !./matmul_single_core
✓ 40s

   ⋯   Benchmarking Matrix Multiplication (M = 2048, N = 512, P = 4096)
       ----------------------Benchmarking Started----------------------
       Naive Matmul: 18.3222 s => 0.468825 GFLOPS
       Transpose (transpose time + multiply time): 3.76355 s (transpose: 0.0205258 s, multiply: 3.74302 s) => 2.29492 GFLOPS (multiply only)
       Loop-Interchange Matmul: 1.61457 s => 5.32026 GFLOPS
       Autotuning block size for matmul_blocked_naive...
          bs=16 -> 2.41077 s
          bs=32 -> 1.23904 s
          bs=48 -> 1.03918 s
          bs=64 -> 0.90421 s
          bs=96 -> 0.806252 s
          bs=128 -> 0.840581 s
       Autotune (naive) picked bs = 96
       Blocked Matmul_Naive (bs=96): 0.821009 s => 10.4627 GFLOPS
       Autotuning block size for matmul_blocked_interchange...
          bs=16 -> 2.5753 s
          bs=32 -> 1.55998 s
          bs=48 -> 1.07221 s
          bs=64 -> 0.930437 s
          bs=96 -> 0.841335 s
          bs=128 -> 0.852971 s
       Autotune (interchanged) picked bs = 96
       Blocked Matmul_Interchanged (bs=96): 0.83489 s => 10.2887 GFLOPS
       All matrix multiplication versions produced correct results (within tolerance).
```

Output Table - 6 Cases-Time

Method ↓ / Case →	64×64 ×64	256×256 ×256	512×512 ×512	1024×1024 ×1024	2048×2048 ×2048	2048×512 ×4096
Naive	200.6 µs	0.0358914 s	0.483542 s	5.85199 s	67.1084 s	18.3222 s
Transpose	188.13 µs	0.0131321 s	0.117796 s	0.990216 s	8.23425 s	3.76355 s
Loop-Interchange	4.576e-05 s	0.00270581 s	0.021956 s	0.167389 s	4.05504 s	1.61457 s
Blocked - Naive	bs=64 5.295e-05 s	bs=96 0.00267472 s	bs=128 0.022495 s	bs=96 0.196648 s	bs=64 1.83544 s	bs=96 0.821009
Blocked – Interchange	bs=128 6.492e-05 s	bs=128 0.00219714 s	bs=128 0.0187048	bs=96 0.157814 s	bs=96 1.69947 s	bs=96 0.83489 s

Output Table - 6 Cases - GFLOP

Method ↓ / Case →	64×64 ×64	256×256 ×256	512×512 ×512	1024×1024 ×1024	2048×2048 ×2048	2048×512 ×4096
Naive	2.6136 GFLOP	0.934887 GFLOP	0.555144 GFLOP	0.366966 GFLOP	0.256002 GFLOP	0.468825 GFLOP
Transpose	3.05903 GFLOP	2.65251 GFLOP	2.32829 GFLOP	0.990216 GFLOP	2.09801 GFLOP	2.29492 GFLOP
Loop-Interchange	11.4573 GFLOP	12.4009 GFLOP	12.2261 GFLOP	12.8293 GFLOP	4.23667 GFLOP	5.32026 GFLOP
Blocked - Naive	9.90157 GFLOP	12.545 GFLOP	11.9331 GFLOP	10.9284 GFLOP	9.36007 GFLOP	10.4627 GFLOP
Blocked – Interchange	8.07591 GFLOP	15.2719 GFLOP	14.3511 GFLOP	13.6077 GFLOP	10.109 GFLOP	10.2887 GFLOP