CS 267 HW 1

Ben Brock

Optimizing Matrix Multiply

- In HW 1, you'll be optimizing **matrix multiply**
- C = C + AB, where A, B, and C are dense matrices
- For simplicity, we'll consider the case of **square matrices**

Problem Pseudocode

```
for i = 1 to N:
  for j = 1 to N:
    for k = 1 to N:
      c[i, j] = c[i, j] + a[i, k] * b[k, j]
```

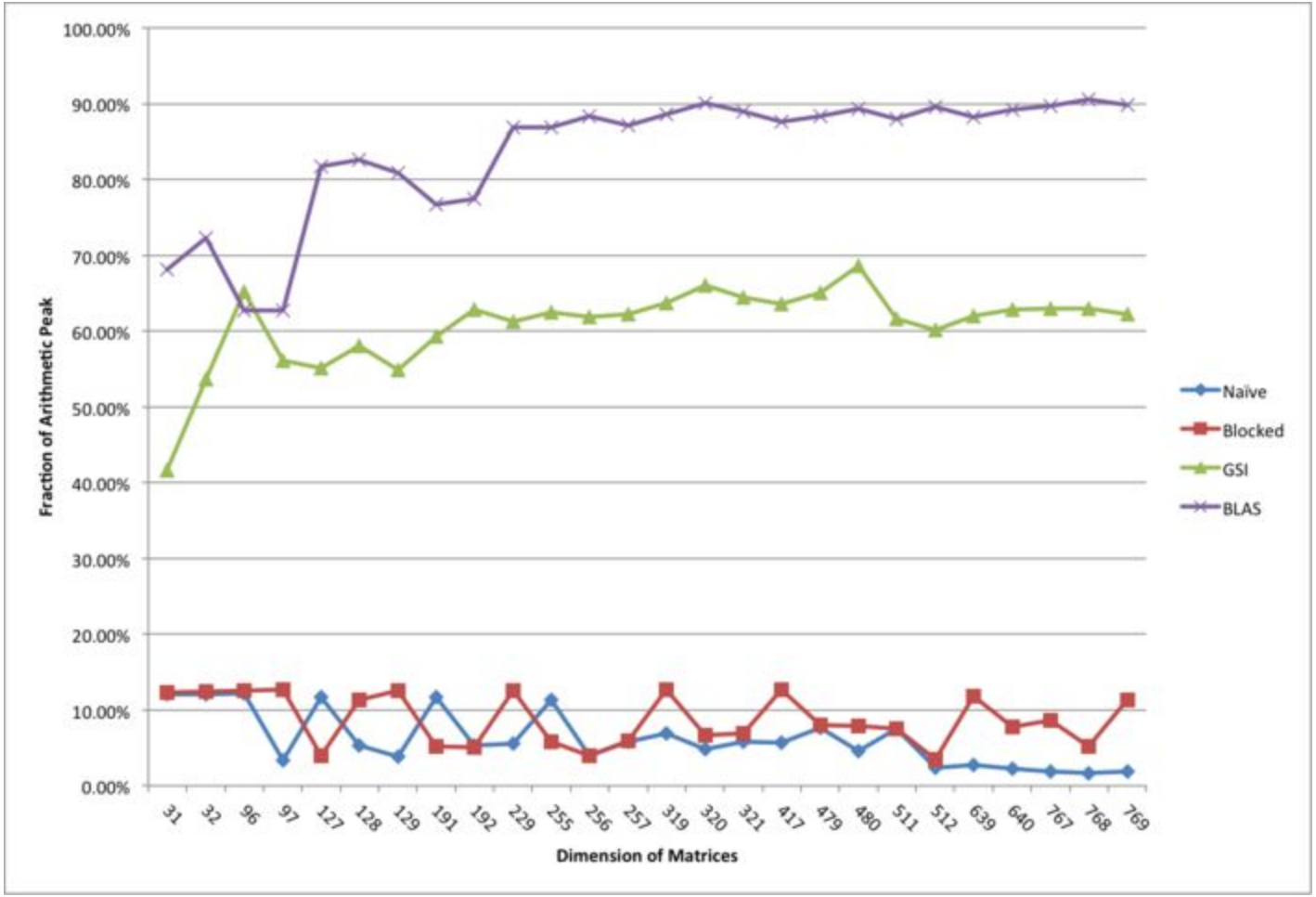
3 nested loops => n^3 complexity

Your Job: Implement This Interface

```
void square_dgemm (int n, double* A, double* B,
                   double* C);
```

You **write** this function, we **call** your function in a test harness.

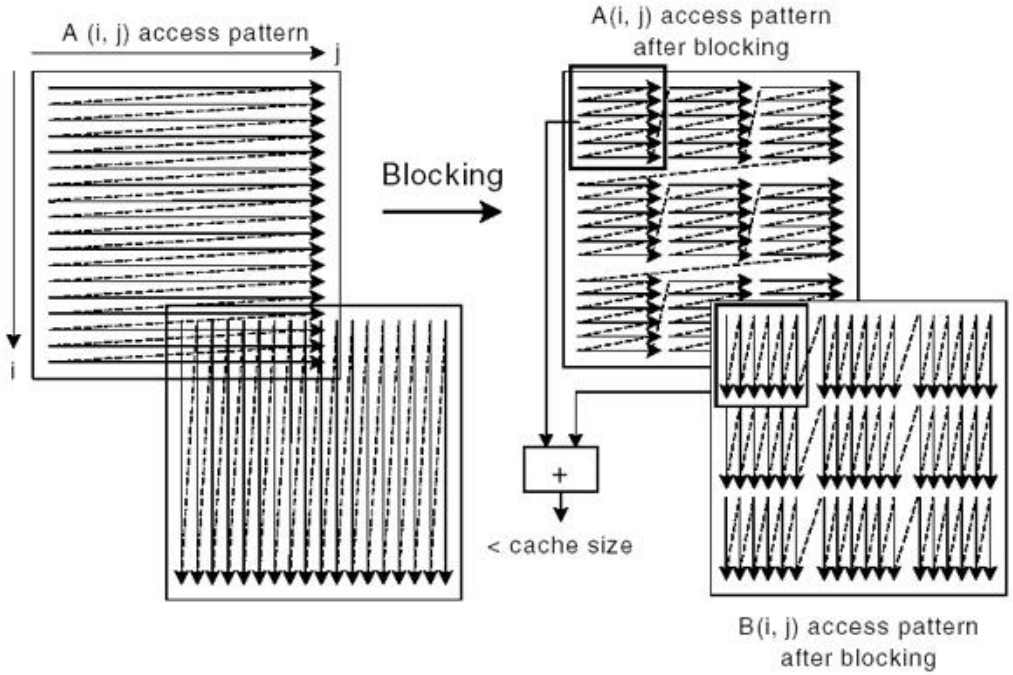
Your job is to make it run as **fast** as possible.


100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
Fraction of Arithmetic Peak
31
32
96
97
127
128
129
191
192
229
255
256
257
319
320
321
417
479
480
511
512
639
640
767
768
769
Dimension of Matrices
Naïve
Blocked
GSI
BLAS

Optimization Techniques

1) Blocking
    a) **L1** blocking
    b) **Register** blocking
    c) **L2** blocking
2) Copy optimization
    a) Copy to an **aligned** buffer
    b) **Transpose**?
3) Vectorization
    a) Write small, fixed-size (n=8-16) GEMM, **examine assembly**
    b) Intrinsics

Blocking (or **Tiling**)



Copy Optimization