assumed: cache size < n
cache line = 8 doubles
only 1 cache

1. Triple loop

\[ \frac{n}{8} \cdot \frac{n}{8} = \frac{n^2}{64} \]

- entry: \( \frac{n}{8} + n \) CNS (compulsory)
- afterwards: \( x \) is in cache
- entry: no reuse, so again \( \frac{n}{8} + n \) CNS

\[ \Rightarrow \text{total} = \left( \frac{n}{8} + n \right) n^2 = \frac{5}{8} n^3 \text{ CNS} \]

2. Blocked

\[ \begin{array}{cccc}
1. \text{block:} & \frac{n^3}{6} + \frac{n^2}{8} = \frac{3n^3}{4} & \text{CNS} \\
2. \text{block: same} & \Rightarrow \text{total} = \frac{n^3}{4} \cdot \left( \frac{n}{8} \right)^2 = \frac{n^3}{64} b \\
\text{choose} & b = \sqrt{\frac{c}{3}} \Rightarrow \frac{\sqrt{3}}{48} n^3 \text{ CNS} \\
\text{gain} & \approx 2.5 \sqrt{c} \\
\end{array} \]

- explains much of triple loop's poor performance (the other major optimization is unrolling and scalar replacement for better instruction parallelism and register usage)
- blocking achieves both: better spatial and better temporal locality with respect to the cache
- in 2.) the number of cache misses = amount of data transferred cache <-> memory is \( O(n^3/\sqrt{c}) \). Hence the operational intensity is \( O(\sqrt{c}) \). It is known that this is optimal, i.e., \( \Theta(\sqrt{c}) \).