Microarchitecture Quiz

How many cycles do the following computations take at least on a Core 2? Data type: double

Example 1: dot product <x,y>

```
t = 0
for i = 0..n-1
    t = t + x[i] * y[i]
return t
```

Based on op count:

$\geq n$ (without vector)

$\geq \frac{n}{2}$ (with vector)

Based on memory:

x,y resident in L1:	$\geq n$
L2:	$\geq 2n$
RAM:	$\geq 8n$

memory is bottleneck

⇒ computation is memory-bound

Example 2: MMM $C = AB + C$

```
for i = 0..n-1
    for j = 0..n-1
        for k = 0..n-1
            $c_{ij} = c_{ij} + a_{ik} b_{kj}$
```

Based on ops:

$\geq n^3$ (no vec)

$\geq \frac{1}{2} n^3$ (vec)

Based on memory:

A,B,C resident in L1:	$\frac{3}{2} n^2$
L2:	$3n^2$
RAM:	$12n^2$

operations are bottleneck

⇒ computation is compute-bound

Important note: The bounds above a all valid, but this very simple analysis does not yet completely settle, whether the computations are really compute or memory bound. In the first case, each vector element is used only once, hence one would expect the performance to be determined by the memory bandwidths. In the second case, however, each element is used n times. Hence every element may need to be loaded several times. It is not easy to see what is the minimum number of such loads. We will discuss this later in the class and see that MMM is indeed compute bound.