EX.2.1.7, Sauer3

Assume that a given computer requires 0.002 seconds to complete backsubstitution on a 4000 upper triangular matrix equation. Estimate the time needed to solve a general system of 9000 equations in 9000 unknowns. Round your answer to the nearest second.

EX.2.1.7, Sauer3, solution, Langou

Colab link: `https://colab.research.google.com/drive/1XR1Czryz1w0nN5sLL8jgzytDjV_NNUYi`

Backsubstitution requires n^2 (floating-point) operations (FLOPs).
If our computer can complete backsubstitution in 0.002 seconds for $n = 4000$, then it performs

```
n = 4000.
gigaflops_per_second = ( n * n ) / 0.002 * 1e-9
print( gigaflops_per_second, 'GigaFLOP per seconds' )
```

```
8.0 GigaFLOP per seconds
```

which means 8 billions operations per seconds. (We say 8 GigaFLOPs/sec.)

Gaussian elimination requires $\frac{2}{3}n^3$ (floating-point) operations (FLOPs). The time, in second, needed for $n = 9000$ is therefore

```
n = 9000.
time_GaussianElimination = \
 ( 2./3. * ( n * n * n ) ) / ( gigaflops_per_second * 1e9 )
print( time_GaussianElimination, 'seconds' )
```

```
60.75 seconds
```

Rounding to the nearest second, we get $\boxed{\text{61 seconds}}$. So about 1 minute.

Not needed for full credit.

We can check that the time for backsubstiution is negligible

```
n = 9000.
time_backsubstitution = ( n * n ) / ( gigaflops_per_second * 1e9 )
print( time_backsubstitution, 'seconds' )
```

```
0.010125 seconds
```

Indeed, the time for backsubstiution is negligible compared to the time for Gaussian elimination. And this makes sense, since backsubstiution is order n^2, and Gaussian elimination is order n^3, and n is 9000. More precisely, the time for backsubstiution is $\frac{2}{3}n$ times smaller than the time for Gaussian elimination, so, with $n = 9000$, it is 6000 times smaller.

Not needed for full credit.

```
n = 9000.
print( f"{2./3. * ( n * n * n ) * 1e-9:4.0f}",\
      'billions of floating-point operations' )
```

```
486 billions of floating-point operations
```

We see that to solve a 9,000 $\times$ 9,000 system of linear of equations, we need to perform (about) 486 billions floating-point operations (FLOPS).