EX.2.2.7, Sauer3

Assume that your computer can solve 1000 problems of type $Ux = c$,where U is an upper-triangular 500-by-500 matrix, per second. Estimate how long it will take to solve a full 5000-by-5000 matrix problem $Ax = b$. Answer in minutes and seconds.

EX.2.2.7, Sauer3, solution, Langou

Colab link: https://colab.research.google.com/drive/13nZp6pUWQOEyIGcnbp8qZUm24686dD7d

Backsubstitution (which is solving $Ux = c$, where U is an upper-triangular) requires n^2 (floating-point) operations (FLOPs).

If our computer can complete 1000 backsubstitutions in 1 second for $n = 500$, then it performs

```
n = 500.
gigaflops_per_second = ( 1000. * ( n * n ) ) / 1. * 1e-9
print( gigaflops_per_second, 'GigaFLOP per seconds' )
```

```
0.25 GigaFLOP per seconds
```

which means 250 millions operations per seconds. (We say 0.25 GigaFLOPs/sec.)

Gaussian elimination requires $\frac{2}{3}n^3$ (floating-point) operations (FLOPs). The time, in second, needed for $n = 5000$ is therefore

```
n = 5000.
time_GaussianElimination = \
 ( 2./3. * ( n * n * n ) ) / ( gigaflops_per_second * 1e9 )
print( time_GaussianElimination, 'seconds' )
```

```
333.3333333333333 seconds
```

So about 5 minutes and 33 seconds.

Note: we neglected the $\mathcal{O}(n^2)$ terms in the solve of the full 5000-by-5000 matrix problem $Ax = b$.