*For the following exercises, one day of work is considered as 8 hours, while one month is considered 24 business days.*

Let’s consider a program that is run every day and lasts for one hour, that is one hour of work is lost every day. Spending a week, that is five days, to improve it will be results in a loss of 4000$, but will improve the speed of the program, which will run in only two minutes each day. That means in approximately 40 days, the wins of time every day will compensate the initial loss of time. So, improving the program in such a case is worth it.

Now, let’s consider a program that is run 1,000,000 times per day, while each run takes one second. Such a case will actually take more than 10 days to be complete. Even after improving it by a factor of 32, such a case still takes more than 8 hours to complete, that is more than a day of work. So, instead of trying to improve such a program, it would be better to try another algorithm, or to revise the experiment.

Finally, let’s consider a program that is run once a month, and take a whole day to complete. If one were to spend a week to increase the speed of the program, this one may run in 15 minutes, instead of the initial 8 hours. This means that each month, 7 hours of work can be saved. However, considering the 4000$ of loss due to improvement on the program, 6 months would be necessary to make this improvement profitable. To conclude, I’d like to say that spending a whole week to improve such a program may be overdoing it. Spending only one or two days may reduce the computation time to 4 up to 2 hours, resulting on a 4 to 6 gained hours by month. In such a case, 2 or 3 months only a required to make profit on the improvement.