1. You have some protein matching code. It completes in 200 hours on your current machine, and spends 20% of the time doing integer operations.

   (a) How much faster must you make the integer unit to make the code run **10 hours faster**?

$$\text{Speedup} = \frac{200}{190} = \frac{1}{\frac{x}{S_{int}} + (1-x)} = \frac{1}{\frac{0.2}{S_{int}} + (1-0.2)}$$

$$\frac{190}{200} = \frac{0.2}{S_{int}} + (1-0.2)$$

$$\frac{190}{200} - 0.8 = \frac{0.2}{S_{int}}$$

$$S_{int} = \frac{0.2}{\frac{190}{200} - 0.8} = \boxed{1.33}$$

   (b) How much faster must you make the integer unit to make the code run **50 hours faster**?

It is impossible, because the max total speedup possible is $\frac{1}{1-0.2} = 1.25$ but the speedup requested by the question is $\frac{200}{150} = 1.\overline{33}$. We can also see this because the above solution for S_{int} gives a negative number when you replace 190 with 150.

2. You have some graph processing code which takes four days to execute on your current machine. Of that time

   - 20% of the time is spent performing integer operations, and
   - 35% of the time is spent performing I/O operations.

   Which of the following is the better tradeoff?

   (a) A compiler optimization that reduces the number of integer instructions by 25% (assume that each integer operation still takes the same amount of time).

   (b) A hardware optimization that reduces the latency of each I/O operation from $6\mu s$ to $5\mu s$.

$$x_{int} = 0.2$$

$$S_{int} = \frac{1}{\frac{0.2}{100/75} + (1-0.2)} = 1.0526$$

$$x_{int} = 0.35$$

$$S_{io} = \frac{1}{\frac{0.35}{6\mu s/5\mu s} + (1-0.35)} = 1.0619$$

Thus, speeding up IO is the better option.

3. Recent advances in process technology have quadrupled the number of transistors you can fit on your processor die. Currently, your key customer can use up to 4 processors for 40% of their application.

   You have two choices:

   (a) Increase the number of processors from 1 to 4.

   (b) Increase the number of processors from 1 to 2, but add features to each processor that will allow the application to use 2 processors for 80% of the execution time.

   Which is the best choice?

$$S_{quad} = \frac{1}{\frac{0.4}{4} + (1 - 0.4)} = 1.4286$$
$$S_{duo} = \frac{1}{\frac{0.8}{2} + (1 - 0.8)} = 1.6667$$

The second option is better!

4. In your application, memory operations currently take 30% of the execution time.

   A new widget called "cache" speeds up 80% of memory operations by a factor of 4.

   A second new widget called "L2 cache" speeds up half of the remaining memory operations by a factor of 2.

   What is the total speedup?

Memory operations are 30% of the execution time.

The L1 cache optimization applies to, 80% of them, so

$$x_{\text{L1}} = 0.3 \times 0.8 = 0.24.$$

The L2 cache optimization applies to 50% of the other 20% of them, so

$$x_{\text{L2}} = 0.3 \times (1 - 0.8) \times (0.5) = 0.03.$$

Also, from the problem, $S_{\text{L1}} = 4$ and $S_{\text{L2}} = 2$. Thus,

$$S_{\text{tot}} = \frac{1}{\frac{x_{\text{L1}}}{S_{\text{L1}}} + \frac{x_{\text{L2}}}{S_{\text{L2}}} + (1 - x_{\text{L1}} - x_{\text{L2}})}$$
$$= \frac{1}{\frac{0.24}{4} + \frac{0.03}{2} + (1 - 0.24 - 0.03)} = 1.2422.$$