The goal of this problem is to decide if we should use a particle-based or a grid-based scheme for calculating the gravitational forces between collections of particles. Ideally, we use whichever one is faster.

If we have n particles, to calculate the force on one particle, we have to sum the forces from all the other particles on it. Since there are n particles, the force on one particle requires n interations. This is $per - particle$, so to get the force on *all* particles, we must do this for each particle, so the total amount of work scales like n^2. The problem specified that the scaling coefficient is a, so $t_{particle} = an^2$. For the FFT-based force calculation, the problem told us to assume we have an m-by-m grid. As we saw in the earlier sample problem, the total amount of work to do a 2-D FFT of this is $m^2 \log(m^2)$, with the scaling constant defined to be b, so $t_{grid} = bm^2 \log(m^2)$. There are other parts to the force calculation, but the problem specified that we should assume those are negligible (which they generally are). The crossover time is when the particle-based execution time and the grid-based execution time is the same, or

$$bm^2 \log(m^2) = an^2$$

The problem asked us to find the critical number of particles at which we should switch to a grid-based scheme, which is:

$$n^2 = \frac{b}{a} m^2 \log(m^2)$$

or

$$n = m\sqrt{\frac{b\log(m^2)}{a}}$$

Note that for 2-D gravity, this just says that whenever we have more than $\sqrt{\frac{b\log(m^2)}{a}}$ particles per column of the mesh we should switch to a grid. For typical values of $a, b, m,$ this will be of order unity, so you should think about switching to a grid when you have more than around one particle per column in the grid.