Problems 1–3 assume the following. You have $n$ coins — they all look identical, and all have the same weight except one, which is heavier than all the rest. You also have a balance scale, on which you can place one set of coins on one side, and another set of coins on the other, and the scale will tell you whether the two sets have the same weight, and if not, which is the heavier set. Assume that $n$ is a power of 3.

1. Devise a strategy that will identify the heavy coin using at most $\log_3 n$ weighings in the worst case.

2. Show that any strategy that correctly identifies the heavy coin must use at least $\log_3 n$ weighings in the worst case.

3. Suppose that every time you weigh two sets of coins, you have to pay $f(k)$ dollars, where $k$ is the total number of coins in the two sets of coins on the scale, and $f$ is some function.
   
   (a) Suppose that $f(k) = \sqrt{k}$. Show how to find the heaviest coin using $O(\sqrt{n})$ dollars.
   
   (b) Suppose that $f(k) = \log_2 k$. Show how to find the heaviest coin using $O((\log n)^2)$ dollars.
   
   (c) Suppose that $f(k) = k^2$. Show how to find the heaviest coin using $O(n)$ dollars.

4. Suppose you have a recursive algorithm that on inputs of size $n > 1$, works as follows: it breaks up the problem into two subproblems, one of size $\lfloor \alpha n \rfloor$, and the other of size $\lfloor \beta n \rfloor$, where $\alpha$ and $\beta$ are positive constants, both of which are less than 1. The time to split the problem into subproblems and combine the solutions is $O(n)$.
   
   (a) Assuming that $\alpha + \beta < 1$, show that the running time is $O(n)$.
   
   (b) Assuming that $\alpha + \beta = 1$, show that the running time is $O(n \log n)$.
   
   (c) Assuming that $\alpha + \beta \leq \min\{1/\alpha, 1/\beta\}$, show that the running time is $O(n^2)$. 