## Code Jam 2008 - AMER Semifinal

# Analysis: Mixing Bowls

Looking at the restrictions on the recipe, it should be obvious that the recipes form a tree. It is also clear that basic ingredients can be ignored: You just throw them into whatever bowl you're currently working on.

Lets consider a recipe with **k** mixtures $\mathbf{m_1}$, ..., $\mathbf{m_k}$. How many bowls do we need to prepare this recipe? It does not make sense to work on more than one mixture at a time, so let's try preparing the mixtures in the given order.

We'll need $\mathbf{b_1}$ bowls to prepare the first mixture. For the second mixture, we can reuse these bowls, except for one bowl that we need to hold the first mixture. While preparing the third mixture (which requires $\mathbf{b_3}$ bowls in itself) we will need two additional bowls to hold the first two mixtures, and so on. Finally, once we have prepared all the mixtures, they sit in **k** different bowls, and we'll need one additional bowl to put everything together, and finish the recipe.

How many bowls did we use in total? Keeping in mind that we can reuse bowls from the previous step (except for the bowls to hold finished mixtures), we will need **b** = max($\mathbf{b_i}$ + **i** - 1) bowls to prepare all mixtures, for a total of max(**b**, **k** + 1) to put everything together.

Looking at this formula, it is obvious that the number of bowls will be minimized if we prepare the mixtures requiring the most bowls first. Thus, our algorithm will first invoke itself recursively for every ingredient, sort the results in descending order, and then use the formula given above to return a result.