2 Egg Drop

(a) $f(1,k) = 1$
f(0, k) = 0
f(n, 1) = n
f(n, 0) = ∞

(b) If we drop eggs from i-th layer $(0 \leq i \leq n)$, two conditions would happen a breath away:
if egg broken, the subproblem will be $f(i-1, k-1)$;
if else, the subproblem will be $f(n-i, k)$.
so the recurrence relation for $f(n,k)$ is:

$$f(n,k) = 1 + min(\{max(f(i-1,k-1), f(n-i,k))\}, i \in \{1,2,...,n\})$$

3 Paper Cutting

(a) We define $B[i_1, j_1, i_2, j_2]$ to be the minimum number of cuts needed to separate the sub-matrix $A[i_1 \leq i_2, j_1 \leq j_2]$ into pieces consisting either entirely of pieces with stains on them or clean pieces.

(b)

$$B[i_1, j_1, i_2, j_2] = \min \begin{cases} 0, \quad \text{if all entries of } A[i_1 \ldots i_2, j_1 \ldots j_2] \text{ are equal} & \\ 1 + B[i_1, j_1, i_1 + k, j_2] + B[i_1 + k + 1, j_1, i_2, j_2] & \text{for any } k \in \{1, \ldots, i_2 - i_1\} \\ 1 + B[i_1, j_1, i_2, j_1 + k] + B[i_1, j_1 + k + 1, i_2, j_2] & \text{for any } k \in \{1, \ldots, j_2 - j_1\} \end{cases}$$

Alternatively, you could have also encapsulated the 0 base case in all single-square pieces, and determined if a piece was pure via the merging, see below.

(c) Two answers are acceptable: $O((m+n)m^2n^2)$ and $O(m^3n^3)$
We have $O(m^2n^2)$ total subproblems: $O(mn)$ possibilities for (i_1, j_1), and $O(mn)$ possibilities for (i_2, j_2). For each subproblem, we examine up to m possible choices for horizontal splits, and n possible choices for vertical splits. A single split consideration will result in two smaller subproblems, which we can assume have already been solved, so we just need to find the best split, which takes $O(n+m)$ time.
In addition, for a subproblem, we also want to check the base case for if the piece is "pure" (contains only clean paper, or contains only paper with stains). Brute force checking this takes $O(mn)$ time, for a total subproblem time of $O(mn + (m+n)) \implies O(mn)$.
However, this $O(mn)$ factor can be reduced to $O(m+n)$ (this is not required to receive full points). We can pre-compute the purities of every single possible subrectangle and store it in a table. Technically, for us to pre-compute faster than $O(m^3n^3)$, we'll need to compute the purities more intelligently than by brute-force. It is possible to do this in as fast as $O(max\{m, n^2\})$ time, although the details of this are complicated and won't be explained here. So to solve our recurrence relation, if we can determine purity/impurity in $O(1)$ time, then we can reach an overall time of $O((m+n)m^2n^2)$.
Alternatively, we can initialize all min-cut values of single square pieces to be 0. Then, if it is possible to have some cut such that both resulting pieces have min-cut values of 0, and both resulting pieces are of the same type (clean-only or stain-only, and we can take any sample of either and compare them), then we ourself are a pure piece. This would allow you to avoid the entire pre-computation business as mentioned before, and still achieve a runtime of $O((m+n)m^2n^2)$.