The problem of eggs and a building

Problem description: You have absolutely identical 2 eggs and empty K-story building. You can throw eggs from any floor and see if it was broken or not. If not, you can reuse it again momentarily. You need to identify the lowest floor, starting from which eggs is broken if thrown ("breaking floor") in minimum possible steps in worst case.

Theorem. For any given integer K you can find out the breaking floor in not more than $\left\lceil\sqrt{2K+\frac{1}{4}}-\frac{1}{2}\right\rceil$ steps (where $\lceil x \rceil$ is the least integers number, equal or greater than x) in worst case scenario.

Proof.

We need to do the following:

I. propose algorithm;
II. make upper bound estimation of number of steps.

We do not intent to proof that the algorithm is optimal in any sense, just providing some bound.

I. Algorithm

Let us fist consider at first three more simple algorithms, and the will get to our more complex and efficient one.

1. Trivial algorithm is then we throw egg starting from floor 1 and up. Clearly we can get K steps in worst case and don't leverage the second egg;
2. The most obvious way of adding second egg is to throw the first egg on each even floor starting from the ground and up, and when it will be broken check previous (odd) floor with second one. This will take K/2 + 1 steps in worst case;
3. The next algorithm is based on observation, that we actually can break K into bigger chunks using the first egg to significantly lower the worst case scenario. Let's make number of chunks equal to chunk size, so that we have $\sqrt{K}$ chunks with size $\sqrt{K}$. Then the worst case is then we go through upper bounds of all (except the last one) chunks with first egg and then in last chunk we go linearly from ground up with second egg. So we have in total $\sqrt{K}-1+\sqrt{K}-1=2\sqrt{K}-2$ steps which is already good but still is worse than we want to get with K big enough.

Our proposed algorithm is based on 3rd one and observation that we want to make chunks less and less each time so that the more steps we do to find right chunks, the less time we need to go through it linearly with the second egg. Obviously, since each next chunk top requires one first egg throw we need each next chunk size to be smaller by one to keep worst-case scenario number of steps the same for each chunk. And we need to ensure that $\sum^{steps} chunk\ size \geq K$. Clearly since we decrease chunk size, the last size can be just 1.

So, we get the following algorithm: if x is number of worst-case steps, we throw the first egg from floor x on step one, then from floor x+(x-1) on second step, then from floor x+(x-1) + (x-2) on third, etc., decreasing chunk size by 1 on each step until the last chunk will be of size 1. If the first egg is broken on some step t<x then the size of last chunk is x-t we use second egg to go linearly from (previous used floor) +1 up to the floor where the first egg was broken, which can take up to x-t steps, so we keep the worst case always t + (x-t) = x steps.

Let me illustrate it on some examples: if building has only 1 floor, then clearly one step is enough, 2 - there 2 clear steps. For 3 floors we have x=2, throwing the first egg at first on floor number 2, the on floor 3 if it was not broken, for 6 floors (and up to 4-floor buildings) we have x=3, throwing on the floor 3, 5, 6, having chunk sizes as 3, 2, and 1 correspondingly on each step.

II. Upper bound

The bound trivially follows from our requirement $\sum^{steps} chunk\ size \geq K$, which means that sum of all chunk sizes on all steps must cover size of the building. As soon as we decreased size of chunks by 1, let us denote x as a number of chunks and we get the following equation for number of steps $\sum_{i=1}^{x} i \geq K$. Therefore, having $\frac{x(x+1)}{2} \geq K$ therefore having $x^2 + x - 2K \geq 0$. As the coefficient with x^2 if greater than 0 that means that the parabola incurved downwards (has a minimum) and as soon as K>0 have 2 roots. Here $D = 1 + 8K$ which means that one of the roots is always negative in our case, as soon as we need only positive one we have $x = \frac{-1+\sqrt{1+8K}}{2} = \sqrt{2K + \frac{1}{4}} - \frac{1}{2}$. **Q.E.D.**

End of the proof.

Example: For K=100 we have $x = \lceil 13.65 \rceil = 14$ so we throw egg from floor 14 first, than from floor 27 = 14 + 13, than from floor 39 = 14 + 13 + 12, than from floors 50, 60, 69, 77, 84, 90, 95, 99, 100, which are the tops of chunks. You can easily check that this will not work for K=100 and x=13 (you will have too much floors left for the last chunk).

Open questions:

- Is there more efficient algorithm?
- What is the optimal algorithms and least number of floors we pass through if we explicitly go up first time and go down and up to get each time it was not broken?
- What is the optimal algorithm and number of throws if we have T identical eggs?
- Joining previous two: what is the optimal algorithms and least number of floors we pass through if we explicitly go up first time and go down and up to get each time it was not broken and we have T identical eggs?

Artem A. Golubev 2009