

2.2 Mecho

Solution 1

Firstly, working with fractional amounts of time is tricky, so we will measure time in units of $\frac{1}{S}$ seconds — let's call them *ticks*. Bees take S ticks to move from one cell to the next, while Mecho takes one tick.

Let us try to solve an easier problem first. Suppose we know when Mecho leaves the honey: can he get home safely? If we can solve this problem, then we can use it inside a binary search to find the last moment at which he can leave.

Mecho's moves will depend on the bees, but the bees' moves are fixed, so let us deal with the bees first. A standard breadth-first search will tell us when the bees reach each grassy cell (this just means simulating the spread of the bees over time).

Next, we can perform a similar breadth-first search for Mecho to answer the question "How soon (if at all) can Mecho reach each cell?" This can be implemented almost exactly as for the bees, except that one must exclude any steps that would have Mecho move to a cell where he would immediately be caught.

These breadth-first searches can each be implemented in $O(N^2)$ time. The problem statement guarantees that Mecho will eventually be caught if he stays with the honey, it takes $O(N^2)$ seconds for the bees to cover all the cells they can reach, and we are only interested in integer numbers of seconds in the binary search. Thus, the range of values explored by the binary search is $O(N^2)$ and hence the time complexity of this solution is $O(N^2 \log N)$.

Solution 2

Instead of using a binary search, we can use a more complicated method to directly determine the optimal time to leave any cell. The bees are processed as in the first solution. However, instead of working from the honey towards Mecho's home, we start from his home. Since he is safe in his home, there is no limit on how late he can arrive there.

Now suppose we know that for some cell Y , Mecho must leave no later than t ticks (from the time the alarm went off) and still make it home safely. If X is a neighbouring cell of Y , what is the latest time Mecho can leave cell X to safely make it home via Y ? Clearly $t - 1$ is an upper bound, otherwise he will reach Y too late. However, he must also leave Y before the bees get there. The latest he can stay will be just the minimum of the two constraints.

One can now do a priority-first search: simulate backwards in time, keeping track of the latest time to leave each cell (keeping in mind that X has other neighbours, and it might be better to leave via those than via Y).

The time complexity of this solution depends on the priority queue used to order cells by leaving time. A binary heap gives an $O(N^2 \log N)$ implementation, and this is sufficient for a full score. However, it can be shown that the number of different priorities that are in the priority queue at any one time is $O(1)$, which makes an $O(N^2)$ solution possible.