



Virus Experiment (Solution)

Solution

Preprocessing the direction of the wind, for each inhabitant, if we know the status (infected or uninfected) of 4 adjacent inhabitants, we can determine whether he or she gets infected or not in $O(1)$ time.

We consider a directed graph whose vertices are inhabitants and edges are the relations “if only this inhabitant is infected, that inhabitant is finally infected.” We decompose it into strong connected components. We need to calculate the minimum number of vertices in a strong connected component of outdegree 0. We also need to calculate the number of strong connected components attaining the minimum number of vertices.

This can be done by the following DFS (depth-first search).

1. We construct a union-find, where we can list all vertices in a union by a merging technique.
2. For a vertex, if every vertex in a union where this vertex belong to is infected, we search the next vertex to be infected.
3. If we arrive at a vertex where we already visited, we stop searching if it is not an ancestor of DFS. If it is an ancestor of DFS, we merge the union of all vertices in the path we followed. (We can determine whether it is an ancestor or not in $O(1)$ time.)

Then, a union means “the set of vertices which infect each other.”

There is a difficulty in speeding up the step 2, especially in listing the vertices adjacent to a union. There are two cases:

- (A) The union consists of one vertex (i.e., we arrive at a vertex for the first time).
- (B) We are considering union which was created by merging two already-existing unions.

The case (A) can be done in $O(1)$ time.

Concerning the case (B), if a vertex is adjacent to only one of two unions to be merged, it was already searched by DFS before merging the unions. Therefore, we need only consider a vertex which is adjacent to both of the two unions. We calculate all vertices which are adjacent to the smaller union. By merging technique, the amortized time complexity is $O(HW \log(HW))$.



Solution (randomized algorithm)

Take a sequence of vertices A where every vertex appears exactly once.

For each of the vertices A_0, A_1, \dots , we do a naive simulation under the condition that it is the only initial infected vertex. In a simulation where A_k is the initial infected vertex, if the vertex A_l is infected and $l < k$, then we quit the simulation.

For simulations which are not quit on the way, we need to calculate the minimum number of vertices which are finally infected, and the number of simulations which attain the minimum.

In this solution, the worst time complexity is $O((HW)^2)$. If we take the initial sequence A randomly, the time complexity becomes roughly $O(HW \log(HW))$.

The sample source file `2019-open-virus-sol1.cpp` implements this solution.

Another solution

For another solution, see the sample source file `2019-open-virus-sol2.cpp`.