## Решење и анализа

Најједноставније решење овог задатка би била чиста симулација по времену. Кратко ћемо описати то решење. За свако поље на мапи чувамо која се бактерија налази на том пољу. У сваком тренутку радимо следеће:

* Проширимо све групе бактерија које се налазе на мапи, тако што на свако поље које додирује неку врсту, упишемо ту врсту.
* Да ли се нека бактерија појављује у тренутку у ком се налазимо? Ако се појављује, убацимо је на мапу тако што обележимо да се на пољу *( X, Y )* налази бактерија врсте *S*.
* Преостаје нам да уништимо неке бактерије. То можемо да урадимо пуштајући *BFS* ( или *DFS* ) од свих поља на којима су се спојиле две групе различите врсте и уклањајући те врсте са мапе.

Једини могући исходи су једна или ниједна преживела. Тренутак након последњег појављивања бактерија, а у ком се на мапи налази само једна врста или уопште нема бактерија је решење. При имплементацији, обратити пажњу да ако се на два суседна поља налази бактерија исте врсте, не мора нужно да значи да та два поља чине исту групу. Анализирајмо сложеност овог решења:

* Проширивање бактерија могуће је имплементирати у сложености *О( maxX ∙ maxY ).*
* Појављивање нове бактерије захтева само њено додавање на одређено поље, па је сложеност *О( 1 ).*
* Уништавање врста може се имплементирати у *О( maxX ∙ maxY ).*

Такмичари који су имали овакво решење освојили су 20 поена.

У симулацији по времену, значајна су само уништавања и спајања група, што можемо искористити за боље решење. На почетку свакој бактерији доделимо чвор у графу и обележимо све бактерије као живе. Све док постоје две живе бактерије различите врсте радимо следеће:

Нађемо две живе бактерије које ће се прве сусрести, уколико постоји више таквих предност дајемо бактеријама исте врсте. Уколико су исте врсте повежемо њихове чворове у графу. Иначе, обележимо обе врсте као мртве и од њихових чворова пустимо *DFS* ( или *BFS* ) и обележимо све бактерије које су повезане са њима ( не нужно директно ) као мртве. Решење је максимум свих тренутака у којима се појављују неке бактерије и свих тренутака у којим долази до уништавања неких бактерија. Уколико постоји нека преживела врста, лако је наћи која је.

Део кода у *C++* који налази растојање између две бактерије описане са *x, y* и *time* као тренутком настанка:

*int get\_dist( node A, node B )*

*{*

*int temp = abs( A.x - B.x ) + abs( A.y - B.y );*

*int diff = max( A.time, B.time ) - min( A.time, B.time );*

*return max( A.time, B.time ) + max( 0, ( temp - diff + 1 ) / 2 );*

*}*

Уништавања може бити највише , међутим спајања у најгорем случају може бити , па је сложеност овог решења *О( N4 )* и оно вреди 50 поена.

Ово решење лако можемо довести до сложености *О( N2 ∙ logN )* која доноси 100 поена. Направимо низ свих парова бактерија. Сви парови ће се некада спојити ( унуштити или повезати ). Сортирајмо овај низ по временима спајања сваког пара, тако што ћемо дати предност паровима који се први спајају, а уколико се два пара спајају у истом тренутку, дајемо предност оном пару који садржи две бактерије исте врсте ( ако такав постоји ). Сада проласком кроз низ радимо исто што и у претходно описаном решењу, само што сада нема потребе да тражимо две најближе бактерије сваки пут. Уколико је бар једна бактерија из пара мртва, прескочимо тај пар, иначе спојимо њихове чворове ако су исте врсте, односно ако су различите врсте уништимо њих и све бактерије повезане са њима.

Напоменимо да и уместо прављења графа и пуштања *DFS*-a ( или *BFS*-a ) могуће је и користити *Disjoint-set (* [*http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=disjointDataStructure*](http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=disjointDataStructure) *)*.