Evacuation

Calvin Fung

August 21, 2025

1 Problem

They've screwed something up yet again... In one nuclear reactor of a research station an uncontrolled reaction is in progress and explosion which will destroy the whole station will happen soon.

The station is represented by a square $n \times n$ divided into 1×1 blocks. Each block is either a reactor or a laboratory. There can be several reactors and exactly one of them will explode soon. The reactors can be considered impassable blocks, but one can move through laboratories. Between any two laboratories, which are in adjacent blocks, there is a corridor. Blocks are considered adjacent if they have a common edge.

In each laboratory there is some number of scientists and some number of rescue capsules. Once the scientist climbs into a capsule, he is considered to be saved. Each capsule has room for not more than one scientist.

The reactor, which is about to explode, is damaged and a toxic coolant trickles from it into the neighboring blocks. The block, which contains the reactor, is considered infected. Every minute the coolant spreads over the laboratories through corridors. If at some moment one of the blocks is infected, then the next minute all the neighboring laboratories also become infected. Once a lab is infected, all the scientists there that are not in rescue capsules die. The coolant does not spread through reactor blocks.

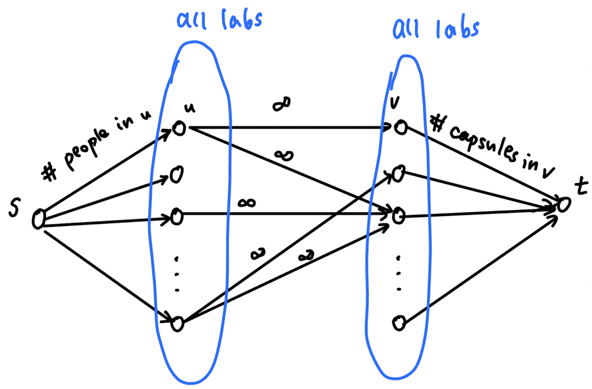
There are exactly $1 \leq t$ minutes to the explosion. Any scientist in a minute can move down the corridor to the next lab, if it is not infected. On any corridor an unlimited number of scientists can simultaneously move in both directions. It is believed that the scientists inside a lab moves without consuming time. Moreover, any scientist could get into the rescue capsule instantly. It is also believed that any scientist at any given moment always has the time to perform their actions (move from the given laboratory into the next one, or climb into the rescue capsule) before the laboratory will be infected.

Find the maximum number of scientists who will be able to escape.

2 Solution

We present a cubic time solution.

Consider the flow network below where there's an edge from lab u to v iff it's possible for a scientist to go from u to v within time t.



It's clear that max flow equal the maximum number of scientists who will manage to save themselves.

Now let t_v be the time that lab v get infected, this can be found by doing BFS from malfunctioning reactor.

Below we show how to figure out all labs v such that it's possible for a scientist to go from s to v within time t. We will achieve this by slightly modifying Dijkstra's algorithm. Let f_v be the earliest time such that it's possible for a scientist to go from s to v. Initially let all $f_v = \infty$ except $f_s = 0$. When we visit lab v and see an unvisited lab u, if $f_v + 1 < t_u$ we update $f_u = \min(f_u, f_v + 1)$ otherwise we don't need to do anything. Finally all labs v such that it's possible for a scientist to go from s to v within time t is just $\{v : f_v \leq t\}$. The proof for correctness is very similar to the classical induction proof [1] for Dijkstra's algorithm, we included in the appendix for completeness.

In conclusion, we can solve this problem by first computing the t values by doing a BFS from malfunctioning reactor in time $O(n^2)$. Then we will run the modified Dijkstra's algorithm for each lab to compute the f values each taking $O(n^2 \lg n)$ time. Last but not least we can construct the flow network based on the f values and the input which will have size $O(n^4)$, finally finding a max flow take $O(n^6)$ time using Push–relabel algorithm with FIFO vertex selection rule [2]. Total runing time is dominated by max flow which is cubic time in terms of size of input $O((n^2)^3)$.

[1] Wiki. `https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm`

[2] Wiki. `https://en.wikipedia.org/wiki/Maximum_flow_problem`

3 Appendix

The proof of correctness for the modified Dijkstra's algorithm :

Invariant hypothesis: For each visited lab v, f_v is the earliest time possible for a scientist to go from s to v, and for each unvisited lab u, f_u is the earliest time possible for a scientist to go from s to u when traveling via visited labs only, or infinity if impossible.

It's easy to check the base case when there is just one visited lab s. Suppose the hypothesis holds for k visited labs, to show it holds for $k + 1$ labs, let u be the next visited lab, i.e. the lab with minimum f_u. The claim is that f_u is the earliest time possible to go from s to u.

The proof is by contradiction. If a shorter path P (such that it's possible for a scientist to go from s to u following this path) were available, then this shorter path either contains another unvisited lab or not. Suppose this P takes $f'_u < f_u$ time.

In the former case, let w be the first unvisited lab on this P. By induction, the earliest time possible for a scientist to go from s to w through visited labs is f_w, this means $f_w < f'_u$. However, $f_u \leq f_w$ otherwise w would have been picked by the priority queue instead of u. This is a contradiction that P is a shorter path.

In the latter case, let w be the last but one lab on P. This means $f_w + 1 \leq f'_u$ and $f_w + 1 < t_u$ (as $f'_u < t_u$). That is a contradiction because by the time w is visited, it should have set f_u to at most $f_w + 1$.

For all other visited lab v, the f_v is already known to be the earliest time possible to go from s to v already, because of the inductive hypothesis, and these values are unchanged.

After processing u, it is still true that for each unvisited lab w, f_w is the earliest time possible to go from s to w using visited nodes only. Any shorter path that did not use u, would already have been found, and if a shorter path used u it would have been updated when processing u.

After all labs are visited, the shortest path from s to any lab v consists only of visited labs. Therefore, f_v is the earliest time possible to go from s to v. ■