Comp 3121 Algorithms & Programming Tech - Question 4

z5335039 Chen En

Question 4 *Spy Escape*

[20 marks] Agency X has sent n spies to Sydney for a secret mission. Before the mission begins, Agency X has prepared $m > n$ secret hideouts throughout the city, of which n of them contain

a single emergency escape pod. The hideouts are connected via a network of tunnels, and each hideout can only accommodate one spy at a time. The emergency escape pods can also only accommodate one spy. Everyday, the spies can crawl through at most one tunnel to reach a new hideout.

The tunnels are represented by an adjacency matrix $T[1..m][1..m]$, where

$$T[i][j] = \begin{cases} \texttt{True} & \text{hideout } i \text{ has a tunnel to hideout } j \\ \texttt{False} & \text{otherwise} \end{cases}$$

4.1 [14 marks] A few days after the mission began, all spies have been compromised. The spies are currently scattered around the hideouts in Sydney and must make it to an emergency escape pod within D days to avoid capture.

Design an $O(nm^2D)$ algorithm which determines whether or not all spies can successfully escape.

4.2 [6 marks] Despite the compromise, Agency X still wants to wrap up some tasks in Sydney before the spies escape. The spies will die if they are not done in D days.

Design an $O(nm^2D \log D)$ algorithm to determine the minimum number of days needed for all spies to successfully escape.

4.1

Construct a flow network with:

1. Source s and sink T.
2. D layers of vertices to present all D days. (layer day1, layer day2, layer day3,.. , layer dayD).
3. For each layer D_k, there are m hideouts. Since all hideouts can only accommodate one spy at a time, we add capacity of 1 for each vertex. We split each hideout into two (in and out vertices) and connect them with an edge of capacity 1. That is, we present the hideout i at day k as $h_{k,i\ (in)} and\ h_{k,i\ (out)}$.
4. For each layer $1 \leq k \leq D$. If T[i][j] = true, we connect hideout i and hide j with a tunnel and present it by constructing an edge of capacity 1 from $h_{k,i\ (out)}$ to $h_{k+1,j\ (in)}$. If a spy remains in the same hideout i, we present it by constructing an edge of capacity 1 from $h_{k,i\ (out)}$ to $h_{k+1,i(in)}$.
5. From source to each $h_{1,i\ (in)}$ is connected by an edge of capacity 1, which presents there is spy in hideout i during day 1. Since there are total n spies in hideouts and each hideout can only accommodate 1 spy, there are total n edges.
6. From each $h_{D,i\ (out)}$ to sink T is connected with an edge of capacity 1 If the hideout i has an emergency escape pod. Since there are n escape pods, there are total n edges.

Consider the flow in this graph:

1. Each $source - h_{1,i\ (in)}$ edge has a capacity of 1, so we present a flow with 1 from S to $h_{1,i\ (in)}$ assigning n spies in hideout during day 1.
2. Each $h_{k,i\ (out)} - h_{k+1,j\ (in)}$. edge has a capacity of 1, if $h_{k+1,j\ (in)}$.receives the flow from $h_{k,i\ (out)}$, which means there is a spy changes the hideout i to hideout j on day k.

To determine whether all the spies successfully escape from day D, we find the maximum flow by applying Ford-Fulkerson's Algorithm to the flow network.

case1: If the maximum flow f = n, which means all spies escape so is possible.
case2: If the maximum flow f < n, which means only f spies escape.

Time complexity

By applying Ford-Fulkerson's Algorithm, the complexity is $O(E|f|)$. The maximum flow is less than or equal to n, and the edge is at most $2n + Dm + Dm^2 < 2m + Dm + Dm^2$. As a result, the time complexity $O(E|f|) = O\big((2n + Dm + Dm^2) * n\big) = \mathrm{O}(nm^2D)$.

4.2
We apply binary search on the day to escape with applying 4.1 Algorithm.
When all the n spies start with the hideout with the escape pod at day 1, it is the minimum day for spies to escape which we set the lower bound to D_1, and we set the upper bound to D_D.
We first find the middle day D_m by calculating the average of the upper bound and lower bound.
To determine whether it is possible for all the spies escape, we find the maximum flow on D_m by applying Ford-Fulkerson's Algorithm to the flow network.

While we do not meet case 3, and there exist day D_i that all spies can escape$(1 \leq i \leq D)$:
case1: If the maximum flow $F = n$, which means the minimum day for spies to escape is on D_m.
case2: If the maximum flow $F < n$, which means only F spies escape on D_i day.
case2.1: if D_m is smaller than the D_i which means i must be between m+1 and the upper bound. So we set the new lower bound to D_{m+1}.
case2.2: if D_m is greater than or equal to D_i and D_{m-1} is also greater than or equal to D_i which means i must be between lower bound and D_{m-1}. So we set the new upper bound to the D_{m-1}.

We run the algorithm again, and search recursively until we meet case 1 or case 3, during that time we keep tighter the bound.

case 3: If both lower bound and upper bound are both D_D. And when we run the algorithm but $F < n$, which means only F spies can escape which means is impossible for spies n to escape in D days.

Time complexity
Since we are using the same method as question 4.1 which the time complexity is $O(nm^2D)$, and applying binary search that takes $O(logn)$. As a result, the time complexity is $O(nm^2DlogD)$.