Comp 3121 Algorithms & Programming Tech - Question 2

z5335039 Chen En

Question 2 *Pirate Gathering*

[30 marks] Somewhere in the Caribbean there are V islands labeled $1..V$ and N pirate ships named $1..N$. Additionally there are E two-way routes, the i^{th} of which allows a ship to sail between island u_i and island v_i. However, the pirates are fiercely territorial and hence each direction of a route can only be used by one pirate ship. That is, if ship x has traveled from island u to island v, another ship y can still travel from island v to island u.

The pirates are initially on unique islands $\{h_1, h_2, \ldots, h_N\}$, and wish to gather on island $T \in [1, ..., V]$.

2.1 **[18 marks]** Design an $O(NE)$ algorithm which determines whether or not it is possible for every pirate to reach the designated island, T.

2.2 **[6 marks]** In an attempt to prevent the gathering, the government has now imposed a restriction that at most s_i pirates are allowed to depart from island i.

Design an $O(NE)$ algorithm which determines whether every pirate can still reach the designated island. You may reference your answer from Question 2.1 and only provide the details and justification of any modifications.

> You may choose to skip Question 2.1 in which case your solution to Question 2.2 will be submitted for both parts.

2.3 **[6 marks]** The location of the designated island T has been leaked to the government and it may no longer be safe. The pirates want to know if there any other options where all pirates can gather, given the same restrictions as Question 2.2. Design an $O(VEN)$ algorithm that counts the number of islands are suitable for the gathering (including island T).

You may reference your answer from Question 2.2 and only provide the details and justification of your modifications, which must also include an updated time complexity justification.

2.1
Construct a flow network with:

1. Construct V vertices for islands, for each island $1 \leq k \leq V$. For N unique islands included in V, for each unique island $1 \leq j \leq V$. And gathering island T is included in V and it also represents the sink T.
2. There are i islands from V allow ships to sail between two islands. If island V_k has a route to island V_m, which $1 \leq m \leq V$. We construct two-ways edges between those two islands that each edge of capacity 1 because a route can only be used by one pirate boat.
3. If island V_k is an isolated island, it does not have any incoming and outgoing edges.
4. For source S has an edge of capacity 1 to each h_j which there is a pirate boat on island j in the beginning. There are total N edges since N boats are sailing to N island.

Consider the flow in this graph:

1. The flow from source S to each h_j shows that each of N boats start with the beginning unique island.
2. The paths from source S to T presents the routes of N boats before all gathering on island T. If there is a flow from V_k to V_m which means a pirate boat sails from island k to m.
3. The maximum flow F ($F \leq N$) presents the number of boats gathering on island T at the end.

To determine whether it is possible for every pirate boat to reach the destination island T, 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 N boats reach the island so is possible.
case2: If the maximum flow F < N, which means only F boats reach the island.

Time complexity

The time complexity can be calculated by $O(E|f|)$. The max flow is bounded by N, and there are at most $(2E + N) < 3E$ edges since every island V contains 2E edges. The time complexity is $O(E|f|) = O((2E + N) * N) = O(2E * N) = O(NE)$.

2.2

Since there is restriction that at most s_i of pirate boats can leave the islands. We construct a flow network similar with 2.1 but adding capacity for some vertices V_i which i islands have routes to other islands:

We first check each island V_k are there routes to connect with other islands, which $1 \leq k \leq V$. If the island V_k connects with other island, we construct an edge of capacity si between $V_{k(in)}$and $V_{k(out)}$. And increment V' which presents the number of islands having routes connected with other islands. There are total V' edges added.
If the island V_k is isolated, we just skip and check the next island.

Consider the flow in this graph:

1. The flow from source S to each $h_{k(in)}$ shows that each of N boats start with the beginning unique island.
2. The paths from source S to T presents the routes of N boats before all gathering on island T. If there is a flow from $V_{k(out)}$ to $V_{m(in)}$ which means a pirate boat sails from island k to m.
3. The maximum flow F ($F \leq N$) presents the number of boats gathering on island T at the end.

To determine whether it is possible for every pirate boat to reach the destination island T, 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 N boats reach the island so is possible.
case2: If the maximum flow F < N, which means only F boats reach the island.

Time complexity

As we first check through all V islands that take O(V) to complete. Then apply the Ford -Fulkerson algorithm, the time complexity can be calculated by $O(E|f|)$. The maximum flow is max flow is bounded by N, and there are at most $2E + N + V' < 4E + N$ edges ($|V'| \leq 2|E|$ ($V' = \{ui \,\&\, vi \; for \; all \; ui \leftrightarrow vi \mid ui \,\&\, vi \; belongs \; to \; V\} \; in \; E \Rightarrow |V'| \leq 2|E|$).
The time complexity is $O(V) + O(E|f|) = O(V + N * (2E + N + V')) = O(V + EN)$.

2.3

Given by the question, island T is no longer safe. Instead of constructing the gathering island T as sink as the previous flow network. Now we have to determine which island V_k is suitable to be the gathering island, so we take V_k as sink to run the algorithm.
We first check if island V_k a good place for gathering, which $1 \leq k \leq V$. If V_k is a good option, we increment G which presents the counts of the islands that are suitable for gathering (case 1 from question2.2). If V_k is not a good option(case 2 from question2.2), we skip it.
After we check through all V islands, G returns the result of the number of gathering islands.

Time complexity

From question 2.2 we get the time complexity of $O(EN)$. Since we run V times of Ford-Fulkerson algorithm by checking V vertices as Sink, the time complexity can be calculated by $O(E|f|) = O(V * (EN)) = O(VEN)$.