6 Thousands Islands

Written by: **Félix Moreno Peñarrubia** (Spain), Universitat Polit'ecnica de Catalunya - BarcelonaTech
Prepared by: Alham Fikri Aji
Solutions, review, and other problem preparations by: Jonathan Irvin Gunawan, Maximilianus Maria Kolbe Lie, Muhammad Ayaz Dzulfikar, Prabowo Djonatan
Analysis author: Maximilianus Maria Kolbe Lie

Definition 6.1. *A canoe whose state is currently docked at island u and can be sailed to island v is defined as (u, v).*

Definition 6.2. *Suppose that canoe i is in the original state, or $(U[i], V[i])$. Define i' as the reversed state (after being sailed for an odd number of times), or $(V[i], U[i])$.*

6.1 Subtask 1

In this subtask, $N = 2$.

Note that each canoe should be sailed an even number of times. For $N = 2$, at least two $(0, 1)$ canoes and one $(1, 0)$ canoe are required for a valid journey. Suppose that $c_1 = c_2 = (0, 1)$ and $c_3 = (1, 0)$. One of the valid journey is $[c_1, c_3, c_2, c'_1, c'_3, c'_2]$.

Time complexity: $O(M)$

6.2 Subtask 2

In this subtask, $N \leq 400$ and the original state of the canoes is guaranteed to be a complete graph.

For $N = 2$, the canoes do not fulfil the minimum requirement for a valid journey, as mentioned in Subtask 1. For $N \geq 3$, a valid journey can be constructed as follows. Suppose that $c_1 = (0, 1)$, $c_2 = (1, 0)$, $c_3 = (0, 2)$, and $c_4 = (2, 0)$. One of the valid journey is $[c_1, c_2, c_3, c_4, c'_2, c'_1, c'_4, c'_3]$.

Time complexity: $O(M)$

6.3 Subtask 3

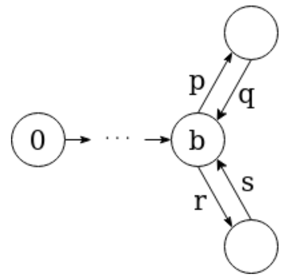
In this subtask, $N \leq 1000$ and the original state of the canoes is guaranteed to be a bidirectional graph.

Observation 6.1. *If an island u has only 1 docked canoe to island v, then a valid journey from island u (to island u) exists if and only if a valid journey from island v (to island v) exists. Therefore, island u can be removed. Instead, find a valid journey from island v.*

Proof. For a canoe $c = (u, v)$, after reaching island u, use canoe c to visit island v. Complete the journey and go back to island v. Use canoe c' to go back to island u. Hence, a valid journey from island u exists. □

First, apply Observation 6.1 starting from island 0 until no island satisfies the observation.

Definition 6.3. *Denote b as the first island from island 0 that does not satisfy Observation 6.1. Refer to the following illustration for more clarity.*



If island b does not exist (the graph is a line graph), then a valid journey does not exist as well. Otherwise, the following is one of the valid journeys from island b. After reaching island b, use the following path: $[p, q, r, s, q', p', s', r']$, which will lead back to island b. By Observation 6.1, a valid journey also exists from island 0. Note that canoe p and r might sail to the same island.

Time complexity: $O(N + M)$

6.4 Subtask 4

In this subtask, $N \leq 1000$ and each edge has a duplicate.

Observation 6.2. *Islands with no docked canoes can be removed without changing the answer. Formally, such islands are nodes with* 0 *out-degree.*

Proof. Using proof by contradiction, if an island with 0 out-degree is part of a valid journey, then the journey will never be a tour, since it is impossible to get out from such an island. □

Note that all canoes which can be sailed to removed islands can be removed as well. The action of removing such islands might result in new islands with no docked canoes, which can be removed too. These operations can be achieved in $O(N + M)$ using DFS until no island can be removed.

The construction in Subtask 3 can be generalized to finding two cycles that can be visited from island b. Some canoes, or none, might be sailed from island b to each cycle. This generalization is correct in Subtask 3, as the edges are bidirectional.

Due to the constraint of this subtask, finding one cycle will automatically find the other cycle. Therefore, a valid journey exists if a cycle can be found in the graph. If no cycle exists in the graph, then island 0 will be removed by Observation 6.2, which implies no valid journey exists. The cycle finding can be achieved in $O(N + M)$ using DFS.

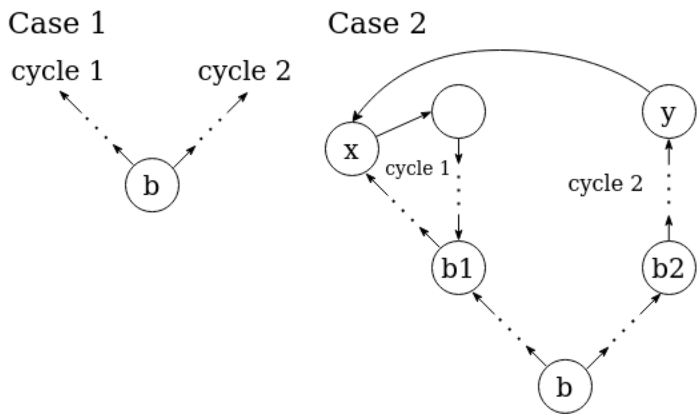
Time complexity: $O(N + M)$

6.5 Subtask 5

To solve the full problem, we shall use some observations and constructions similar to those from the previous subtasks. However, the algorithm that removes islands in Observation 6.1 and 6.2 should be done simultaneously. Keep removing the islands until no island satisfies either observation.

Next, two valid cycles should be found from island b. Note that the cycle refers to a set of canoes, not a set of islands. It is possible to visit the same islands, as long as it uses different canoes.

After reaching island b, there are 2 different cases that need to be handled. These cases are easier to be described by the following illustrations. Note that the dots in the illustration refer to a list of canoes, which can consist of zero or more canoes.



Case 1: all canoes used in both cycles are disjoint. Using the similar construction in Subtask 4, a valid journey exists in this case.

Case 2: there exist the same canoes in both cycles. The following construction satisfies this case. After reaching island b, go to island b_1. Finish the first cycle, then go back from island b_1 to island b. Go to island b_2 and traverse the second cycle until island x, which is a part of the first cycle. Finish traversing the first cycle in reverse from island x. Then, from island x, go back to island y, and finish traversing the second cycle in reverse until island b. Two cycles are completed, which implies a valid journey exists in this case.

Time complexity: $O(N + M)$