



# I can guess the data structure

Link submit:

[https://uva.onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&page=show\\_problem&problem=3146](https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=3146)

Solution:

C++	<a href="https://ideone.com/FDMiZM">https://ideone.com/FDMiZM</a>
Java	<a href="https://ideone.com/AQdVkd">https://ideone.com/AQdVkd</a>
Python	<a href="https://ideone.com/btLijH">https://ideone.com/btLijH</a>

Tóm tắt đề:

Có một cái túi giống như cấu trúc dữ liệu.

Cho một dãy thao tác, đoán xem đó là cấu trúc dữ liệu gì. Nó có thể là một stack (vào sau – ra trước), queue (vào trước – ra trước), priority queue (phần tử lớn nhất ra trước) hay một cái gì đó khác bạn khó có thể tưởng tượng. Có hai loại thao tác:

- Loại 1: đưa một phần tử có giá trị  $x$  vào trong cấu trúc dữ liệu.
- Loại 2: lấy phần tử đầu tiên của cấu trúc dữ liệu đó ra.

Input:

Gồm nhiều bộ test case. Mỗi test case gồm:

Dòng đầu tiên chứa một số nguyên dương  $n$  là số lượng thao tác ( $1 \leq n \leq 1000$ ).

$n$  dòng tiếp theo, mỗi dòng chứa hai số type,  $x$  – lần lượt là loại thao tác, giá trị tương ứng thao tác đó ( $1 \leq x \leq 100$ ).

- type = 1: đưa giá trị  $x$  vào cấu trúc dữ liệu.
- type = 2: lấy giá trị đầu tiên của cấu trúc dữ liệu ra không bị lỗi và nó phải bằng  $x$ .

Output:

Với mỗi test case, in ra cấu trúc dữ liệu trong test case đó đang ám chỉ stack, queue hay priority queue. Hoặc là impossible nếu như cấu trúc dữ liệu này không thể là loại nào trong 3 loại trên, hoặc not sure nếu như cấu trúc dữ liệu đó có thể là nhiều hơn 1 trong 3 cấu trúc dữ liệu đề cập ở trên.

**Ví dụ:**

6	queue
1 1	not sure
1 2	impossible
1 3	stack
2 1	priority queue
2 2	
2 3	
6	
1 1	
1 2	
1 3	
2 3	
2 2	
2 1	
2	
1 1	
2 2	
4	
1 2	
1 1	
2 1	
2 2	
7	
1 2	
1 5	
1 1	
1 3	
2 5	
1 4	
2 4	

**Giải thích ví dụ:**

Ví dụ gồm 5 bộ test.

**Bộ 1:** đẩy vào cấu trúc các số lần lượt là 1, 2, 3, lấy ra lần lượt là 1, 2, 3. Các số vào trước ra trước, vậy đây là queue.

**Bộ 2:** đẩy vào cấu trúc các số lần lượt là 1, 2, 3, lấy ra lần lượt là 3, 2, 1. Các số vào sau ra trước, đồng thời các số lớn nhất được lấy ra trước, vậy đây có thể là stack hoặc priority queue. Nên đáp án là not sure.

**Bộ 3:** đẩy vào cấu trúc số 1, lấy ra số 2. Lấy phần tử đầu tiên trong cấu trúc ra là 1, không như mong đợi (2). Vậy đây không thể là 1 trong 3 cấu trúc đã đề cập, nên kết quả là impossible.

Các bộ còn lại tương tự.

### Hướng dẫn giải:

Bài này ta chỉ việc đi mô phỏng lại các thao tác ứng với ba cấu trúc được đề cập và kiểm tra.

Đầu tiên, chuẩn bị 3 cờ isStack, isQueue và isPQ có giá trị là 1/0 nhằm thể hiện rằng cấu trúc dữ liệu hiện tại của mình có phải là stack, queue hay priority queue không. Ban đầu chưa biết cấu trúc đó là gì thì ta khởi tạo 3 cờ này bằng 1 hết.

Tiếp theo, khởi tạo stack, queue, priority queue là rỗng.

Với lệnh type = 1, ta đẩy x vào stack, queue, priority queue.

Với lệnh type = 2, đây là lệnh yêu cầu ta lấy 1 phần tử ra khỏi cấu trúc dữ liệu đó, nên trước khi thực hiện thao tác này, kiểm tra stack, queue, priority queue có rỗng hay không. Nếu như chúng rỗng mà thao tác này lại bảo mình lấy 1 phần tử ra, thì rõ ràng không thể tồn tại cấu trúc dữ liệu này. Do đó ta gán isStack = isQueue = isPQ = 0.

Ngược lại, nếu chúng không rỗng, ta kiểm tra phần tử mà ta lấy ra có bằng x hay không. Nếu phần tử đầu tiên của stack mà khác x thì cập nhật isStack = 0. Làm tương tự với queue và priority queue.

Cuối cùng, ta gọi total là số lượng cờ đúng. Nếu total > 1, ta in not sure. Nếu total = 0, ta in ra impossible, ngược lại thì kiểm tra cờ của stack, queue hay priority queue bằng 1 thì trả về cấu trúc dữ liệu tương ứng.

**Độ phức tạp:**  $O(N\log N)$  với N là số lượng thao tác. Độ phức tạp sẽ đạt cao nhất khi thao tác trên priority queue.