2.5.1 Homework

Created by TUNG DUC NGUYEN tung2.nguyen, last modified 40 minutes ago

Bài 1: DIGOKEYS - Find the Treasure (medium)

https://www.spoj.com/problems/DIGOKEYS/

Có N hộp đã bị khóa được đánh số từ 1 đến N trừ hộp thứ nhất. Hộp thứ N chứa kho báu, N-1 hộp còn lại chứa m chìa khóa để mở m hộp có vị trí tương ứng. Mỗi lần mở chỉ được mở 1 hộp. Bắt đầu với hộp số 1 không bị khóa. Tìm cách để có thể mở được hộp N với số hộp cần mở là nhỏ nhất. Nếu không có cách nào mở được hộp, in ra -1.

Input:

Dòng đầu tiên T là tổng số test case.

Ứng với mỗi test case sẽ có input như sau:

Dòng thứ nhất là số N là tổng số hộp.

N-1 dòng tiếp theo, ở dòng thứ *ith* có số M là tổng số chìa khóa bên trong hộp này. M số tiếp theo của dòng này là Mj tương ứng là chìa khóa có thể mở được hộp thứ *jth.*

Output:

Với mỗi test case in ra số q là số hộp cần mở.

Dòng tiếp theo bao gồm q số tương ứng là các hộp cần mở. Nếu có nhiều cách để mở, in ra cách có thứ tự mở hộp theo từ điển là nhỏ nhất.

Điều kiện:

1 <= T <= 10

2 <= N <= 100000

1 <= M <= 10

Ví dụ:

1

5

1 4

1 5

1 5

2 2 3

Ví dụ này có 1 test case. Tổng số có 3 hộp. Hộp 1 chứa 1 chìa khóa có thể mở được hộp 4. Hộp 2 chứa 1 chìa khóa mở được hộp 5. Hộp 3 chứa 1 chìa khóa mở được hộp 5. Hộp 4 chứa 2 chìa khóa mở được hộp 2 và 3. Có 2 cách mở hộp 5 như sau:

Cách 1: Mở hộp 1 -> mở hộp 4 -> mở hộp 2 -> mở hộp 5. Các hộp cần mở để tới hộp 5 {1, 4, 2}

Cách 2: Mở hộp 1 -> mở hộp 4 -> mở hộp 3 -> mở hộp 5. Các hộp cần mở để tới hộp 5 {1, 4, 3}

Cả 2 cách đều có thể mở hộp 5 sau 3 lần mở hộp nhưng cách 1 là đáp án đúng vì {1, 4, 2} được xem là nhỏ hơn khi sắp xếp các cách theo thứ tự từ điển.

Bài 2: Monk and the Magical Candy Bags (medium)

https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/monk-and-the-magical-candy-bags/

Monk có N cái túi đựng kẹo. Túi thứ *ith* chứa Ai viên kẹo. Sau khi nhặt một túi lên và ăn tất cả các viên kẹo trong đấy, túi sẽ tự đầy lại kẹo nhưng số lượng chỉ bằng 1 nửa so với số kẹo ban đầu. Ví dụ nhặt túi ith lên và ăn hết Ai viên kẹo trong đó, sau khi ăn hết, số kẹo trong túi ith sẽ tự hồi lại [Ai/2] viên kẹo (làm tròn xuống). Monk phải về nhà trong K phút nữa. Trong mỗi phút Monk chỉ có thể ăn hết số kẹo trong 1 túi. Xác định tối đa số kẹo Monk có thể ăn.

Input:

Dòng đầu tiên là T tương ứng là tổng số test case.

Trong mỗi test case có 2 dòng:

Dòng đầu có 2 số N và K là tổng số túi Monk có và số phút Monk phải về nhà.

Dòng tiếp theo gồm N số tương ứng là tổng số kẹo của N túi.

Output:

In ra tổng số kẹo tối đa mà Monk có thể ăn.

Điều kiện:

1 ≤ T ≤ 10

1 ≤ N ≤ 10^5

0 ≤ K ≤ 10^5

0 ≤ Ai ≤ 10^10

Bài 3: CLEANRBT - Cleaning Robot (medium)

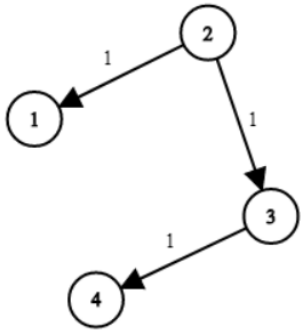
https://www.spoj.com/problems/CLEANRBT/

- Một sàn nhà có kích thước WxH (W,H <=20). Trên sàn nhà có K ô gạch bị bẩn (K<=10).
- Sàn nhà được đánh dấu bằng các ký tự '.' (robot đi qua đường) và 'x' (robot không đi qua được).
- Robot có thể di chuyển nhiều lần qua cùng một ô.
- Vị trí ban đầu của robot được đánh dấu là 'o'; Vị trí viên gạch bẩn được đánh dấu '*'
- Robot chỉ có thể di chuyển theo hướng trên/dưới/trái/phải.
- Tìm số bước di chuyển nhỏ nhất của robot để làm sạch hết các ô gạch bị bẩn, hoặc in ra -1 nếu không thể làm sạch hết các ô gạch này.

~~Bài 4: Network Delay Time (medium)~~ <use Dijkstra instead of BFS>

~~https://leetcode.com/problems/network-delay-time/~~

~~Cho một mạng lưới N nodes, đánh số từ 1 đến N, và một mảng (u, v, w) biểu thị thời gian truyền từ node u đến node v mất w thời gian. Bắt đầu phát tín hiệu từ node k, trả về thời gian để tất cả các node nhận được tín hiệu, nếu không trả về -1.~~




```
Input: times = [[2,1,1],[2,3,1],[3,4,1]], n = 4, k = 2
Output: 2
```

~~Giới hạn: 0 <= N <= 100, 1 <= times.length <= 6000~~

~~Bài 5: Top K Frequent Elements (medium)~~ <Not clear meaning of Heap usage>

~~https://leetcode.com/problems/top-k-frequent-elements/~~

~~Cho mảng A và số nguyên k, tìm ra k phần tử xuất hiện nhiều nhất trong mảng A.~~

~~Ví dụ: A = [1,1,1,2,2,3], k = 2~~

~~Output: [1, 2]. Số 1 và số 2 là 2 số có số lần xuất hiện nhiều nhất trong mảng A~~

⭐⭐⭐: Problems and Explanations are contributed by @HUNG TRONG HO hung.ho @HUY QUANG LE huy2.le @DUC NHAT NGUYEN duc4.nguyen @NAM TIEN NGUYEN nam4.nguyen

No labels