



# KEFA AND PARK

Link submit: <http://codeforces.com/problemset/problem/580/C>

Solution:

C++	<a href="https://ideone.com/AVZuz2">https://ideone.com/AVZuz2</a>
Java	<a href="https://ideone.com/JaqlnT">https://ideone.com/JaqlnT</a>
Python	<a href="https://ideone.com/Yp6FFY">https://ideone.com/Yp6FFY</a>

**Tóm tắt đề:**

Kefa quyết định ăn mừng nhân dịp lần đầu anh ta nhận được lương khủng bằng cách đi đến nhà hàng.

Anh ấy sống ở một công viên khác thường. Công viên là một gốc cây gồm  $n$  đỉnh, với đỉnh gốc là 1, đó cũng là nơi chứa nhà của Kefa. Công viên có nhiều mèo. Kefa tìm ra các đỉnh chứa những con mèo.

Các đỉnh lá của công viên chứa nhà hàng. Kefa muốn chọn một nhà hàng sẽ đi nhưng không may là anh ấy rất sợ mèo, vì thế sẽ không đời nào anh ấy đi tới nhà hàng nếu đường đi từ nhà hàng đến nhà của anh ấy chứa nhiều hơn  $m$  đỉnh liên tiếp chứa mèo.

Nhiệm vụ của bạn là giúp Kefa đếm số nhà hàng anh ấy có thể đi.

**Input:**

Dòng đầu tiên chứa 2 số nguyên  $n$  và  $m$  ( $2 \leq n \leq 10^5, 1 \leq m \leq n$ ) – số đỉnh của cây và số đỉnh chứa mèo liên tiếp tối đa vẫn ổn với Kefa.

Dòng thứ hai chứa  $n$  số  $a_1, a_2, \dots, a_n$ , mỗi  $a_i$  có thể bằng 0 (tức là đỉnh  $i$  không có mèo) hoặc 1 (nếu đỉnh  $i$  chứa mèo).

$n - 1$  dòng tiếp theo chứa cạnh của cây theo định dạng " $x_i y_i$ " (không có ngoặc) ( $1 \leq x_i, y_i \leq n, x_i \neq y_i$ ), với  $x_i$  và  $y_i$  là đỉnh của cây nối nhau bằng một cạnh.

Dữ liệu đảm bảo rằng tập các cạnh là của 1 cây cụ thể.

**Output:**

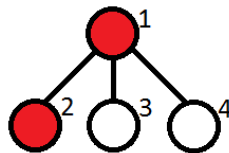
Một số nguyên – là số lá phân biệt của cây có đường đến nhà Kefa chứa tối đa  $m$  đỉnh liên tiếp có mèo.

### Ví dụ:

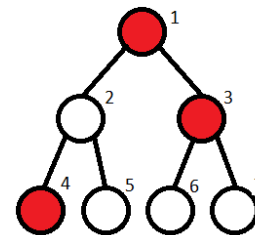
4 1 1 1 0 0 1 2 1 3 1 4	2
7 1 1 0 1 1 0 0 0 1 2 1 3 2 4 2 5 3 6 3 7	2

### Giải thích ví dụ:

**Ví dụ 1:** Các đỉnh chứa mèo đánh dấu màu đỏ. Nhà hàng nằm ở 2, 3, 4. Kefa chỉ không thể đi đến nhà hàng ở đỉnh 2.



**Ví dụ 2:** Nhà hàng nằm ở 4, 5, 6, 7. Kefa không thể đi tới nhà hàng 6, 7.



### Hướng dẫn giải:

Với bài này, ta phải thử đi hết các đường đi từ nút gốc (nhà Kefa) đến tất cả nút lá (nhà hàng) để xem có bao nhiêu nút lá có thể đến được. Vậy ta sẽ sử dụng BFS để giải quyết bài này.

Số đỉnh đánh dấu từ 1 nên khi lưu trữ dữ liệu và chạy BFS, ta cũng bắt đầu từ 1. Dùng một mảng `cat`, `cat[i]` ứng với số lượng mèo xuất hiện khi đi từ đỉnh 1 đến đỉnh `i`. Và dùng một biến để đếm số nút lá (nhà hàng) có thể đến được. Xét nếu đỉnh 1 chứa mèo thì tăng `cat[1]` lên 1. Kefa chỉ đi đến nhà hàng khi đường đó không xuất hiện quá `m` con mèo, nên ta chỉ cần xét những đỉnh mà số lượng mèo xuất hiện nhỏ hơn hoặc bằng `m`, lớn hơn `m` không cần xét nữa. Vậy khi chạy BFS, thực hiện xét các đỉnh kề với đỉnh đang xét `u`, nếu đỉnh `v` chưa thăm thì đánh dấu đã thăm. Đồng thời, nếu đỉnh đó chứa mèo thì cập nhật lại `cat[v] = cat[u] + 1`, và nếu `cat[v]` nhỏ hơn hoặc bằng `m` thì kiểm tra nếu số đỉnh kề với `v` là 1 (đồng nghĩa với việc chỉ có 1 đỉnh đi đến đỉnh `v` thì `v` là nút lá) thì tăng biến đếm lên, ngược lại thì đẩy đỉnh `v` vào hàng đợi.

**Độ phức tạp:**  $O(V + E)$  với  $V$  và  $E$  lần lượt là số lượng đỉnh và cạnh của đồ thị.