Competitive Programming SS24

Submit until end of contest

Problem: idol (1.0 second timelimit)

Ruby and Aqua have finally started to make their debuts in the show world. But at this rate it's going to take years for them to become popular enough to take their revenge. To speed the process up, they have decided they need to eliminate some of their competition.

They have already hacked into the audition system for a famous television show and obtained another participant's application. It turns out to be a .cpp file attemping to solve the famous problem *chase*[1]. To solve *chase* one must find the highest possible median edge weight of a walk with length at least k. The program seems to follow the basic structure of the solution presented in a lecture they saw once.

Now they just need to prove that it's wrong (for differing values of k) by finding some counterexamples. Attached is the submission file. Can you help the twins?

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    int n, m, k;
    cin >> n >> m >> k;
    vector<tuple<int,int,int>> edges(m);
    for (auto& [u, v, w] : edges)
        cin >> u >> v >> w, --u, --v;
    // Binary Search for the first q that returns false.
    cout << *ranges::partition_point(views::iota(0,1e6+1), [&] (int q) {
        vector dp(2*k-1, vector<int>(n, -1e9));
        fill(begin(dp[0]), end(dp[0]), 0);
        // dp[i][v] = Maximum count of edges with w >= q on a path to v with length i or negative if no such path exists
        for (int i = 1; i < 2*k-1; i++) {
            for (const auto& [u, v, w] : edges)
                dp[i][v] = max(dp[i][v], dp[i-1][u] + (w >= q));
            if (i >= k && *max_element(begin(dp[i]), end(dp[i])) > i/2)
                return true;
        }
        return false;
    }) - 1;

    return 0;
}
```

[1] It might be useful to reference the original problem and solution notes.

Input The input contains a single **even** integer k ($2 \leq k \leq 30$), a parameter of the input provided to the above solution.

Output On the first line, output two integers n, m ($1 \leq n \leq 5 \cdot 10^4, 1 \leq m \leq 10^5$), the number of nodes and edges in your counterexample respectively.

On each of the following m lines you should print three integers u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^6$), representing a directed edge from the node u_i to v_i with weight w_i.

Note that u_i may equal v_i, but for any $i \neq j$ it must hold that $u_i \neq u_j$ or $v_i \neq v_j$. This means that your graph may contain self loops, but the same edge shall not appear twice, even with different weights.

Your output will be augmented with the value k from the input and given into the provided program. Your answer is correct if and only if the program then returns a wrong answer.

It is guaranteed that such a counterexample exists for all values k that satisfy the constraints.

There are no samples for this special task.