Assignment 5 : CSL 201
Topic : Red-Blue Spanning Trees

This assignment is based on Exercise 28, Chapter 4 in the book "Algorithm Design" by Kleinberg and Tardos. Your objective is to implement a polynomial time algorithm for the following problem:

INPUT. A connected, undirected graph $G(V, E)$. Each edge $e \in E$ is colored either *red* or *blue*. We are also given an integer k, $0 \leq k \leq n-1$.

OUTPUT. Answer "Yes" if there exists a spanning tree of G with exactly k red edges and $n-1-k$ blue edges. Otherwise, answer "No". In case your answer is positive, also output the description of a spanning tree with k red and $n-1-k$ blue edges.

1 Input Format

The first line of the input contains a single number k : the number of red edges required in the spanning tree. The next line contain a single number n, the number of vertices in graph G. There are n more lines after that - the ith line describes the adjacency list of vertex i.

Suppose the the degree of vertex i in G is d_i. Then, the ith line contains a sequence $v_1 b_1 v_2 b_2 \ldots v_{d_i} b_{d_i}$ of $2d_i$ integers. Here $v_1, v_2, \ldots, v_{d_i}$ are the neighbours of vertex i in graph G, and $b_j, 1 \leq j \leq d_i$, is either 0 or 1 depending on whether the color of edge $(i, v_j) \in E$ is red or blue. A -1 at the end of each line denotes the end of the adjacency list for the corresponding vertex.

2 Output Format

On the first line of output, your program will print either "Yes" or "No", based on whether a spanning tree with the given number of red edges exists in G or not. If the answer is "Yes", the algorithm will also print the description of *one* spanning tree T of G with exactly k red edges. This description will be provided on $n-1$ separate lines, with the ith line describing the endpoints of ith edge in tree T.

Note that there may exist several spanning trees with exactly k red edges, and hence the output of the algorithm is not unique. You will get full credit as long as you output the description of one such spanning tree.

3 Example

Consider the red-blue undirected graph given in Figure 1.

Suppose this graph is given as input with $k = 4$:

```
4  // the number k of red edges
9  // the number of vertices in the graph
```

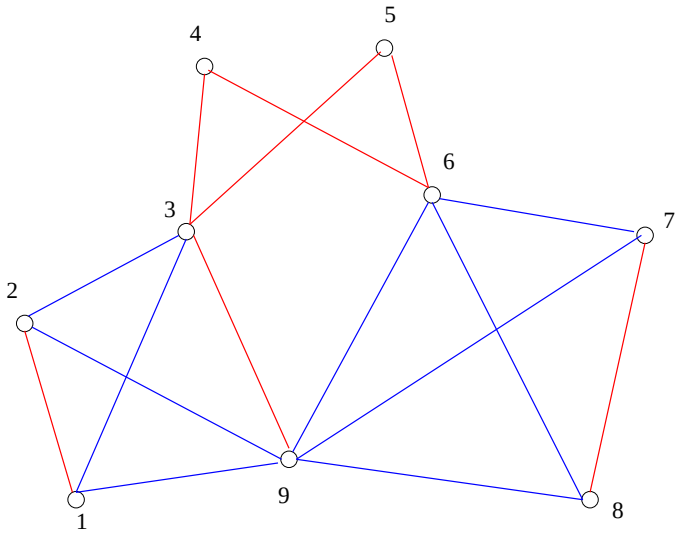

Figure 1: Red-blue connected, undirected graph

```
9 1 2 0 3 1 -1
1 0 9 1 3 1 -1
2 1 1 1 9 0 4 0 5 0 -1
3 0 6 0 -1
3 0 6 0 -1
7 1 8 1 9 1 4 0 5 0 -1
6 1 9 1 8 0 -1
7 0 6 1 9 1 -1
1 1 2 1 3 0 6 1 7 1 8 1 -1
```

One possible correct output for your program on the above input is:

```
Yes
3 4
2 3
1 3
9 3
3 5
4 6
6 8
6 7
```

The program outputs "Yes" and outputs the $n - 1 = 8$ edges of a spanning tree T with exactly 4 red edges. One can verify that the program will output "Yes" for $k = 2, 3, 4, 5, 6$ and "No" for $k = 0, 1, 7, 8$.

For now, write the code for solving the red-blue spanning tree problem and test it on your own examples. We will provide you with test cases in due time.