Problem B. Hamu

Time limit:	1 second
Memory limit:	1024 megabytes

Dingdong plans to travel to the country of Hamu.

The country of Hamu consists of n cities and m bidirectional roads. Before this trip, Dingdong has already visited city i exactly a_i times. During this trip, Dingdong plans to enter Hamu at city s. On each subsequent day, Dingdong will travel along a certain road to a city and visit that city once. At the end of the final day's visit, Dingdong should be at city s and leave Hamu from city s. **Note that on the day of entering the country of Hamu, Dingdong does not visit city s.**

Dingdong hopes that after this trip, combined with his previous visits, every city in Hamu will have been visited by him an even number of times. Dingdong has limited time and can visit at most $5n$ cities during this trip. Please help him construct a valid trip plan, or tell him if it is impossible.

Input

The input contains multiple testcases. The first line of the input contains an integer T ($1 \le T \le 2 \times 10^5$), the number of testcases.

For each testcase, the first line contains three integers n, m, s ($1 \le n \le 2 \times 10^5, 0 \le m \le 2 \times 10^5, 1 \le s \le n$), where n is the number of cities, m is the number of roads, s is the index of the starting city.

The next line contains n integers $a_1, a_2, \cdots, a_n$ ($0 \le a_i \le 10^9$), the number of times Dingdong has visited for each city.

The next m lines each contain two integers u_i, v_i ($1 \le u_i, v_i \le n$), indicating an undirected road connecting city u_i and city v_i. **There can be multiple edges and self loops.** In other words, it's **not** guaranteed that $u_i \neq v_i$, and it's **not** guaranteed that $(u_i, v_i) \neq (u_j, v_j)$ for $i \neq j$.

It's guaranteed that the sum of n and the sum of m over all testcases does not exceed 2×10^5, respectively.

Output

For each testcase, if there's no valid trip plan, print `No` in a single line.

Otherwise, print `Yes` in a single line first, then print an integer k ($0 \le k \le 5n$) in the next line, representing the total number of visits during this trip. In the following line, print the indexes of the cities visited in order.

Example

standard input	standard output
5 4 4 1 1 1 1 1 1 2 2 3 3 4 4 1 5 7 1 9 4 3 11 7 1 2 2 5 2 4 3 4 2 3 1 3 4 5 2 2 2 114 514 1 2 1 2 5 0 1 114 514 19 19 810 2 1 1 3 5 1 2	Yes 4 2 3 4 1 Yes 6 2 4 5 2 3 1 Yes 0 No Yes 2 2 1

Note

For the first testcase, after visiting along the route $1 \to 2 \to 3 \to 4 \to 1$, cities $1, 2, 3, 4$ have been visited $2, 2, 2, 2$ times respectively, meeting the requirement.