Algorithms and Complexity - 1st Assignment

Translated to English

1 - Statistical Alchemies

In the country of Algorithms, there are recent concerns regarding the availability of cereal products during winter. To alleviate these concerns and make new friends, you gather data about the availability of cereal products in N major supermarkets along the central commercial road of the capital. It is known that in the storage facilities of supermarket i, there are c_i available packages. It is observed that in some cases, the available packages are sufficient, while in others, they are very few and likely insufficient for winter.

You do not want to draw conclusions from extreme cases, whether the availability is too high or too low. After careful consideration, you conclude that a good estimate of the availability of cereal products is derived from the **median** of the available packages across at least K supermarkets, for an appropriately chosen value of K. To address the issue optimistically, you accept that the maximum value of these medians is a representative estimate of the cereal product availability, and you want to write a program to calculate it.

Input

The program initially reads from the standard input two positive integers: the number N of supermarkets for which you know the availability of cereal products and the minimum number K of consecutive supermarkets you will use for your estimation. On the next line, N positive integers $c_1, c_2, \ldots, c_N$ are given, representing the availability of cereal products in each supermarket.

Output

The program must print a single positive integer, which represents the maximum median that can result from considering at least K consecutive supermarkets. Recall that the median of a set of numbers is the value at position $\lceil (K+1)/2 \rceil$ after sorting them in ascending order.

Constraints:	**Input Examples:**	**Output Examples:**
$1 \leq K \leq N \leq 2 \cdot 10^5$ $1 \leq c_i \leq N$ Runtime limit: 1 sec. Memory limit: 64 MB.	5 3 1 2 3 2 1 4 2 1 2 3 4 10 2 1 10 2 6 10 8 9 4 4 5	2 3 9

2 - Spanning Trees with 2 Criteria

Recently, an important discussion has arisen in the country of Algorithms about the usefulness and efficiency of various algorithmic techniques. In their attempt to resolve these discussions, the country's scientific advisors are preparing to present their conclusions to the President of the country. The President is trying to balance perspectives and explain that all techniques are useful and that solving complex algorithmic problems often requires a combination of algorithmic techniques. For instance, it is suggested to compute a spanning tree of a graph that maximizes the ratio of the total profit to the total weight of the edges it includes.

More specifically, we consider a directed connected graph $G(V, F)$ with $|V| = N$ vertices and $|E| = M$ edges. Each edge e offers profit $\rho(e)$ and imposes weight $w(e)$ if it is included in the selected spanning tree

T of G. The task is to calculate a spanning tree T' of G that maximizes the ratio $\sum_{e\in T'} \rho(e)/\sum_{e\in T'} w(e)$. The President claims that solving this problem requires a combination of algorithmic techniques and asks for a program to implement this algorithm.

Input:

The program will initially read from the standard input two positive integers: the number N of vertices and the number M of edges of the connected graph. Each of the next M lines will contain four positive integers $u(e), v(e), \rho(e), w(e)$ that represent an edge e. The two integers $u(e)$ and $v(e)$ denote the end-points of edge e, and it is guaranteed that $u(e) \neq v(e)$. The integers $\rho(e)$ and $w(e)$ denote the profit and weight of edge e.

Output:

The program must print to the standard output two integers: the total profit $p(T) = \sum_{e\in T} \rho(e)$ and the total weight $w(T) = \sum_{e\in T} w(e)$ of the spanning tree T of G that maximizes the ratio $p(T)/w(T)$. Specifically, the program must output the integers $p(T)/gcd(p(T), w(T))$ and $w(T)/gcd(p(T), w(T))$, separated by a space (the gcd operation will handle cases where there are multiple optimal spanning trees).

Constraints:	**Input Examples:**	**Output Examples:**
$2 \leq N \leq 50,000$ $1 \leq M \leq 200,000$ $1 \leq u(e), v(e) \leq N$ $1 \leq \rho(e), w(e) \leq 200$	3 3 1 2 1 3 2 3 2 2 3 1 3 1	5 3
Runtime limit: 3 sec. Memory limit: 64 MB.	4 5 1 2 2 3 2 3 3 1 3 4 1 2 4 1 2 1 2 4 4 4	3 2