National Technical University
School of Electrical and Computer Engineering Department of Information Technology and Computers **Algorithms and Complexity** Teachers: Art. Pagourtzis, D. Fotakis, D. Souliu, P. Grondas **1st Series of Programming Exercises - Delivery Date 11/13/2023**

Exercise 1: Statistical Alchemies

In the land of Algorithms there are recent concerns about the sufficiency of grain products during the winter. In an effort to entertain these concerns, and thereby make new friends, you have gathered data on the availability of cereal products in the N mega-supermarkets along the capital's main shopping street. So you know that ci packages are available in the warehouses of supermarket i . You find that in some cases the packs available are sufficient, although they never exceed N, while in other cases, they are very few and probably not enough for the winter.

You don't want to draw conclusions from extreme cases, with high or low availability. After some thought, you conclude that a good estimate of the availability of cereal products comes from the median of packages available in **at least** K consecutive supermarkets, for an appropriately chosen price of K. In your attempt to be optimistic, you accept that the maximum value of these medians is a representative estimate of the availability of grain products, and you want to write a program that calculates it. **Input Data:** Your program will initially read from standard input two positive integers, the number N of supermarkets for

which you know the availability of cereal products, and the minimum number of consecutive supermarkets K to consider for your estimation. In the next line, N positive integers c1, . . . , cN separated by a space between them. The integer ci corresponds to the availability of packages of cereal products of supermarket i. **Output Data:** Your program should print to standard output a positive integer, representing the maximum through value that can be achieved on a segment of the central trade street with **at least**

K consecutive supermarket locations. Recall that the median of a sequence of K numbers is the value at position ÿ(K + 1)/2ÿ of the corresponding sorted (in ascending order) sequence.

Restrictions:

1 ÿ K ÿ N ÿ 2 · 105 1 ÿ ci ÿ N

Execution time limit: 1 sec.
Memory limit: 64 MB.

Input Examples:	**Output Examples:**
5 3 1 2 3 2 1	2
4 2 1 2 3 4	3
10 2 1 10 2 6 10 8 9 4 4 5	9

Exercise 2: Bi-Criteria Splicing Trees

Controversy has recently erupted in the land of Algorithms regarding the usefulness and efficiency of various algorithmic techniques. Among the more intransigent are Greed advocates and Binary Search advocates. The President of the country is trying to calm them down spirits and to explain that all techniques are useful and that efficient solving of complex algorithmic problems usually requires a combination of algorithmic techniques. As an example, he suggests computing a spanning tree that maximizes the ratio of the total cost to the total weight of the edges it contains.

More specifically, we consider an undirected connected graph G(V, E) with |V | = N vertices and |E| = M edges. Each edge e offers a profit p(e) and incurs a weight w(e) if included in the chosen spanning tree of G. The goal is to compute a spanning tree T of G w(e). The President of the country claims that solving this problem efficiently requires which maximizes the ratio ÿ $_{eÿT}$ p(e)/ÿ $_{eÿT}$ a clever combination of algorithmic techniques and asks you to write a program that confirms this claim.

Output Data: Your program will initially read from standard input two positive integers, the count N of vertices and the count M of edges of a connected graph. In each of the following
M lines, will be given four positive integers u(e), v(e), p(e), w(e) representing an edge e. The first two integers denote the vertices u(e) and v(e), with u(e) ÿ= v(e), which are the edges of e. The next two integers denote the gain p(e) and the weight w(e) of edge e.

Output Data: Your program should print to standard output two integers, the total gain p(T) = ÿ maximizes the ratio p(T)/w(T). To be $_{eÿT}$ p(e) and the total weight w(T) = ÿ $_{eÿT}$ w(e) of the spanning tree T of G which precise, your program should print the integers p(T)/gcd(p(T), w(T)) and w(T)/gcd(p(T), w(T)), separated by a gap between them (division by NQD deals with the case where there is more than one optimal spanning tree).

Restrictions:

2 ÿ N ÿ 50,000 1
ÿ ÿ 200,000 1 ÿ
u(e), v(e) ÿ N 1 ÿ
p(e), w(e) ÿ 200 For
60% of the score, it will be w(e)
= 1 .

Execution time limit: 3 sec.
Memory limit: 64 MB.

Input Examples:	**Output Examples:**

3 3
1 2 1 3
2 3 2 2
3 1 3 1
	5 3

4 5
1 2 2 3
2 3 3 1
3 4 1 2
4 1 2 1
2 4 4 4
	3 2