A. Scheduling latency guarantee in 5G network slicing

time limit per test: 2 minutes
memory limit per test: 1024 megabytes
input:standard input
output:standard output

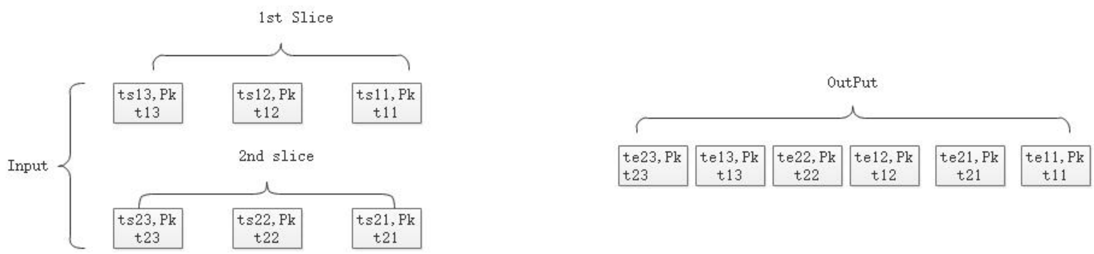
Background

With the emergence of diversified new services in the 5G and cloud era, different industries, services, and users have various Quality of Service(QOS) requirements on networks. 5G network slicing is a network architecture that enables the multiplexing of virtualized and independent logical networks on the same physical network infrastructure. Each network slice is an isolated end-to-end network tailored to fulfill diverse requirements requested by a particular application. Deterministic latency guarantee is an important feature of network slicing. User traffic is transmitted in packets on the network. When multiple slices share one egress, the sequence of inter-slice packet scheduling affects the packet latency. Soft slicing uses scheduling algorithms to schedule packets of different slices, thereby implementing latency isolation between slices.

Task

An objective of the task is to design a soft slice scheduling algorithm. By properly arranging the scheduling sequence between slices, as many slices as possible meet the bandwidth and delay constraints, and at the same time, the maximum delay waiting for all slice packet scheduling is minimized.

The following figure shows the scheduling result example in which two slices are scheduled by using the round robin (RR) scheduling algorithm. where $Pkt_{i,j}$ indicates the jth packet in the ith slice， $te_{i,j}$ indicates the leave time of the first bit of $Pkt_{i,j}$, and $ts_{i,j}$ indicates the arrival time of the last bit of $Pkt_{i,j}$. In the scheduling algorithm the scheduling granularity is at least one packet.



The goal of a task can be modeled as:

$$\max\left(\sum_i f_i/\mathrm{n} + \frac{10000}{\max_i(Delay_i)}\right)$$

$$f_i = \begin{cases} 1, Delay_i \le UBD_i \\ 0, Delay_i > UBD_i \end{cases}$$

$$Delay_i = \max_j\,(te_{ij} - ts_{ij})$$

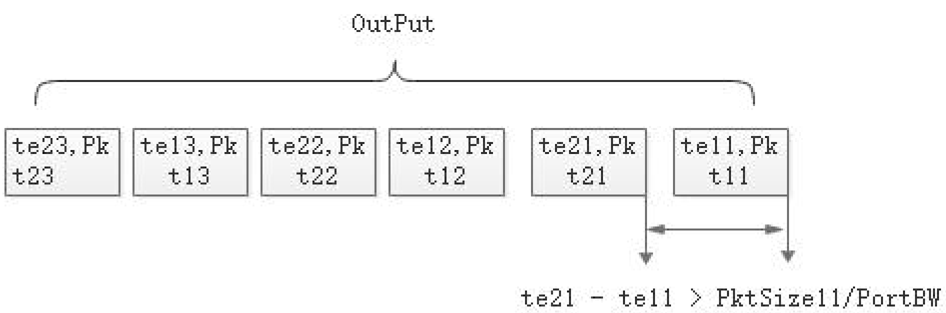
where f_i indicates whether the maximum scheduling waiting delay of the ith slice ($Delay_i$) exceeds the maximum delay tolerable by the slice(UBD_i) (which is a slice attribute and is provided in the input). If $Delay_i$ does not exceed UBD_i, then $f_i = 1$,else $f_i = 0$. $Delay_i$ is a maximum value of the leave time minus the arrival time of all packets in the ith slice. n indicates the total number of slice users.

The scheduling algorithm must meet the following constraints:

1, The scheduling output sequence must meet the port bandwidth constraint(PortBW). Assume that the leave time of the previous packet is $te_{i,j}$, the size of the dequeued packet is $PktSize_{i,j}$, and the leave time of the next packet is $te_{k,m}$, the following requirements must be met.

$$te_{k,m} - te_{i,j} \ge \frac{PktSize_{i,j}}{PortBW}$$

PortBW is a constant ranges from 1 Gbps to 800 Gbps. $PktSize_{i,j}$ is the packet size of the jth packet in the ith slice. The size of the input packet in the slice is variable, and the value ranges from 512 bit to 76800 bit. The timestamp of packet($te_{i,j}$ and $ts_{i,j}$) must be an integer, which unit is ns.



2, Output bandwidth constraint for the ith slice ($\mathrm{SliceBW_i}$): Assume that the arrival time of the first packet in the ith slice is $ts_{i,1}$ and the departure time of the last packet in the ith slice is $te_{i,m}$. The following conditions must be met.

$$\frac{\sum_j PktSize_{i,j}}{te_{i,m} - ts_{i,1}} \ge 95\% * SliceBW_i$$

$SliceBW_i$ ranges from 0.01Gbps to 10Gbps.

3, Packet scheduling sequence constraint: Packets in the same slice must leave in the order in which they arrived, and the packet leaving time must be longer than the packet arrival time.

$$te_{i,j+1} \ge te_{i,j}$$
$$te_{ij} \ge ts_{ij}$$

4, Constraints on scheduling algorithm implementation: During the actual scheduling algorithm decision-making process, the arrival sequence of future packets cannot be predicted. To ensure the algorithm effect, only the information of the arrived packets can be used for each scheduling decision. In addition, all test cases must use the same scheduling policy. It is not allowed to use multiple scheduling policies to obtain the highest score and output the result. This constraint needs to be manually determined.

Input

Line 1: Number of slice users n(1 < n <= 10000), PortBW(Gbps)
Line 2: number of first slice packets m_1, slice bandwidth $SliceBW_1$, and maximum slice delay tolerance UBD_1
Line 3: sequence information about the first slice
$ts_{1,1}\ PktSize_{1,1}\ ts_{1,2}\ PktSize_{1,2} ... ts_{1,m1}\ PktSize_{1,m1}$
...
Line 2n: number of slice packets m_n, slice bandwidth $SliceBW_n$, and maximum slice delay tolerance UBD_n
Line 2n+1: sequence information of the nth slice
$ts_{n,1}\ PktSize_{n,1}\ ts_{n,2}\ PktSize_{n,2}$ ... $ts_{n,mn}\ PktSize_{n,mn}$

Output

Line 1: Total number of scheduled packets

Line 2: the output scheduling sequence is as follows:

te0 SliceId0 PktId0 te1 SliceId1 PktId1 ... teK SliceIdK PktIdK

Note: Your program should use the system's standard input stream and standard output stream for input and output. The Output Slice ID and Packet ID must start from 0.

Scoring

The goal of the task is to minimize the maximum scheduling delay of sliced packets. Therefore, Score $=\sum_i f_i/\text{n} + \frac{10000}{\max_i(Delay_i)}$. If the output does not meet the constraint conditions, the test case score is 0.
The total score of a test set is the sum of the scores of all test cases.

Example

input

```
2 2
3 1 30000
0 8000 1000 16000 3000 8000
3 1 30000
0 8000 1000 16000 3000 8000
```

output

```
6
0 0 0 4000 1 0 8000 0 1 16000 1 1 24000 0 2 28000 1 2
```

The maximum waiting delay is calculated as follows: $te_{2,3}$-$ts_{2,3}$ **= 28000 - 3000 = 25000. Score = 2/2 + 10000/25000 = 1.4**