

2.3 Обработка сетевых пакетов

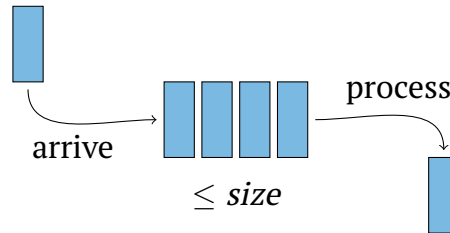
Обработка сетевых пакетов

Реализовать обработчик сетевых пакетов.

Вход. Размер буфера $size$ и число пакетов n , а также две последовательности $arrival_1, \dots, arrival_n$ и $duration_1, \dots, duration_n$, обозначающих время поступления и длительность обработки n пакетов.

Выход. Для каждого из данных n пакетов необходимо вывести время начала его обработки или -1 , если пакет не был обработан (это происходит в случае, когда пакет поступает в момент, когда в буфере компьютера уже находится $size$ пакетов).

Ваша цель — реализовать симулятор обработки сетевых пакетов. Для i -го пакета известно время его поступления $arrival_i$, а также время $duration_i$, необходимое на его обработку. В вашем распоряжении имеется один процессор, который обрабатывает пакеты в порядке их поступления. Если процессор начинает обрабатывать пакет i (что занимает время $duration_i$), он не прерывается и не останавливается до тех пор, пока не обработает пакет.



У компьютера, обрабатывающего пакеты, имеется сетевой буфер размера $size$. До начала обработки пакеты хранятся в буфере. Если буфер полностью заполнен в момент поступления пакета (есть $size$ пакетов, поступивших ранее, которые до сих пор не обработаны), этот пакет отбрасывается и уже не будет обработан. Если несколько пакетов поступает в одно и то же время, они все будут сперва сохранены в буфер (несколько последних из них могут быть отброшены, если буфер заполнится).

Компьютер обрабатывает пакеты в порядке их поступления. Он начинает обрабатывать следующий пакет из буфера сразу после того, как обработает текущий пакет. Компьютер может простаивать, если

все пакеты уже обработаны и в буфере нет пакетов. Пакет освобождает место в буфере сразу же, как компьютер заканчивает его обработку.

Формат входа. Первая строка входа содержит размер буфера $size$ и число пакетов n . Каждая из следующих n строк содержит два числа: время $arrival_i$ прибытия i -го пакета и время $duration_i$, необходимое на его обработку. Гарантируется, что $arrival_1 \leq arrival_2 \leq \dots \leq arrival_n$. При этом может оказаться, что $arrival_{i-1} = arrival_i$. В таком случае считаем, что пакет $i - 1$ поступил раньше пакета i .

Формат выхода. Для каждого из n пакетов выведите время, когда процессор начал его обрабатывать, или -1 , если пакет был отброшен.

Ограничения. Все числа во входе целые. $1 \leq size \leq 10^5$; $0 \leq n \leq 10^5$; $0 \leq arrival_i \leq 10^6$; $0 \leq duration_i \leq 10^3$; $arrival_i \leq arrival_{i+1}$ для всех $1 \leq i \leq n - 1$.

Пример.

Вход:

```
1 0
```

Выход:

Если пакетов нет, выводить ничего не нужно.

Пример.

Вход:

```
1 1
0 0
```

Выход:

```
0
```

Пакет поступил в момент времени 0, и компьютер тут же начал его обрабатывать.

Пример.

Вход:

```
1 2
0 1
0 1
```

Выход:

```
0
-1
```

Первый пакет поступил в момент времени 0, второй пакет поступил также в момент времени 0, но был отброшен, поскольку буфер в этот момент полностью заполнен (первым пакетом). Первый пакет начал обрабатываться в момент времени 0, второй был отброшен.

Пример.

Вход:

```
1 2
0 1
1 1
```

Выход:

```
0
1
```