

## Третье задание по практикуму для самостоятельного программирования дома

Напишите программу, формирующую дерево процессов заданной топологии. Направленный граф-дерево задаётся так же, как и во втором задании, списком смежности, которому предшествует число вершин  $N$ . Вершины пронумерованы в порядке обхода дерева «в глубину», корень дерева имеет номер 1, в списке смежности вершины идут по возрастанию. В ходе построения дерева процессов должно быть выведено  $N$  строк: в строке с номером  $k$  сначала должен быть выведен PID процесса, соответствующего  $k$ -й вершине, а затем последовательность PID его сыновей, расположенных в порядке, соответствующем списку смежности.

Чтобы описать то, что происходит после того как дерево *целиком* построено, предположим, что у каждого процесса есть некоторое множество целых чисел. Тогда каждый процесс производит следующие действия:

1. Получает по одному числу от каждого из своих сыновей.
2. Берёт своё наименьшее число, которое ещё не рассмотрено. Если числа закончились, то выбирается ноль.
3. Сумму чисел сыновей и своего числа передаёт отцовскому процессу.

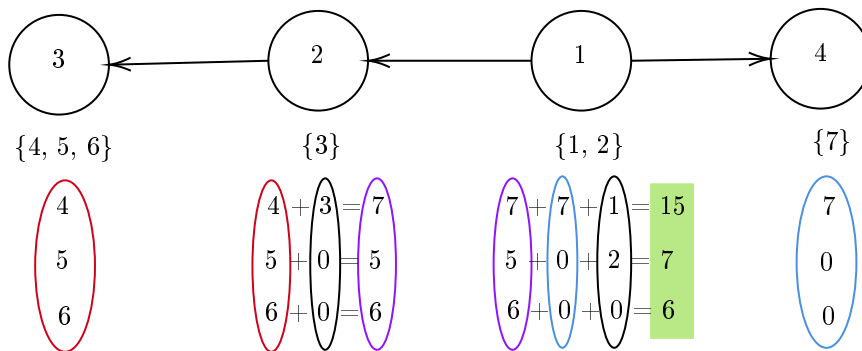
Процессы-листья не выполняют первый пункт, корневой процесс не выполняет последний пункт. В результате указанных действий числа к корневому процессу будут приходить «волнами». Суммы чисел в каждой из таких «волн» в порядке их прихода корневой процесс должен запомнить, чтобы вывести их в конце.

Множества чисел, которыми обладают процессы, задаются сразу после списка смежности, по одному множеству в строке. Каждая такая строка начинается с количества чисел, которыми обладает очередной процесс, а далее следуют сами числа. При этом процессы идут в том же порядке, в котором они шли в списке смежности.

### Пример ввода:

```
4
2 2 4
1 3
0
0
2 1 2
1 3
3 4 5 6
1 7
```

Получающееся дерево и множества чисел каждой вершины изображены на рисунке.



Ниже множеств чисел на рисунке изображены арифметические действия, которые будет производить каждый из процессов на протяжении трёх числовых «волн». После того как процесс закончит производить арифметические операции, он должен завершить свою работу, предварительно написав строку вида «Process <PID> finished». После того как все остальные процессы завершат свою работу, корневой процесс должен вывести список из ранее запомненных сумм, написать строку о своём завершении и завершить свою работу.

Таким образом, **полный вывод для примера** в предположении о том, что вершине с номером  $k$  соответствует процесс с PID, равным  $(k + 1000)$ , получится примерно следующим:

```
1001 1002 1004
1002 1003
1003
1004
Process 1003 finished
Process 1002 finished
Process 1004 finished
15 7 6
Process 1001 finished
```

### Требования:

1. Единственным допустимым средством межпроцессного взаимодействия являются неименованные каналы, связывающие процесс с его сыном. *Не должно быть* передачи информации по каналам между процессами, не связанными родственными отношением отец-сын, например, между процессом и его внуком. Для каждой пары отец-сын разрешается заводить *не более* двух каналов.
2. Строку с информацией о том, в какие вершины идут рёбра от  $k$ -й вершины, должен прочитать процесс, соответствующий вершине  $k$ , и только он. Сразу же как только эта строка будет считана, т. е. сразу как пользователь нажмёт Enter, сыновьи процессы этого процесса должны быть порождены, и очередная строка с PID процесса и его сыновей должна быть выведена. Таким образом, для примера выше содержимое терминала должно выглядеть примерно таким образом:

```
4
2 2 4
1001 1002 1004
1 3
1002 1003
0
1003
0
1004
2 1 2
1 3
3 4 5 6
Process 1003 finished
Process 1002 finished
1 7
Process 1004 finished
15 7 6
Process 1001 finished
```

3. Каждый процесс должен считать с терминала *только свой* список чисел.
4. Вывод на терминал должен производиться атомарно, выводы разных процессов не должны смешиваться.
5. Завершение работы процесса должно происходить только после завершения работы всех его сыновей.

**Мягкий срок сдачи: 25.11.2023.**

**Жёсткий срок сдачи: 23:59, 1.12.2023.**