

- A. Testrun
- B. kolakoski
- C. necklace
- D. rocks

E. shipping

Contest Analysis

Questions asked 2

Submissions

Testrun	
0pt	Not attempted 0/6 users correct (0%)
kolakoski	
8pt	Not attempted 5/7 users correct (71%)
17pt	Not attempted 2/5 users correct (40%)
necklace	
16pt	Not attempted 10/10 users correct (100%)
29pt	Not attempted 9/10 users correct (90%)
rocks	
7pt	Not attempted 2/2 users correct (100%)
53pt	Not attempted 0/1 users correct (0%)
shipping	
26pt	Not attempted 2/6 users correct (33%)
44pt	Not attempted 0/1 users correct (0%)

Top Scores

bmerry	103
Marcin.Smulewicz	71
shik	70
MiSawa	60
ZbanIlya	53
WJMZBMR	45
simonlindholm	45
mk.al13n	45
wan92hy	45
dreamoon	24

Problem E. shipping

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

small 26 points 2 minute timeout	The contest is finished.
large 44 points 10 minute timeout	The contest is finished.

Problem

In a certain country that shall remain unnamed, people live in **N** villages along river valleys separated by mountain ranges. Due to the geography of the country, it is extremely difficult to cross over mountains. The only practical way to get from one village to another is by following rivers. All of the rivers are part of the same system and each river connects to another river downstream, until eventually all water in the system passes through a single point to enter the sea. There is a village at every point where two rivers join and at each river's source, and there is one village at the mouth of the entire river system, where it meets the sea. There may be also be villages at other points along rivers. No two villages are at the same location along a river, and there is exactly one path between any two villages. In other words, the villages and the paths between them form a tree.

You are in charge of managing a shipping company that delivers packages between villages by transporting them on boats along the rivers. Since there is only one path between any two villages, you fortunately do not have to worry about finding the shortest path to use. Unfortunately, however, the country is in the middle of a civil war, with **K** rival factions battling for control. Each village is under the control of exactly one of the factions. Luckily, conditions are stable at the moment and no village is going to change factions anytime soon.

As a neutral company, you are able to send your boats through every village, but only under the condition that each of the factions is allowed to use your shipping services for free, by loading additional packages onto any of your boats that pass between villages under the control of that faction. Every time one of your boats passes through a village (including the village at which your boat begins its route), you may be given an additional package occupying 1 unit of capacity on your boat, to be transported to another village further along on your boat's current route that is occupied by the same faction. Because the factions do not want their packages to be intercepted by other factions, you will be given a package to transport between two villages only if every village on the path between them (including the destination) is occupied by the same faction. Once you deliver a package, you can reuse the space for another package later on, but multiple packages at the same time require multiple units of capacity.

You are now faced with the problem of guaranteeing enough extra capacity on your boats to transport these extra packages. You have **Q** shipments, each of which has a source village and a destination village. You will use a different boat for each shipment. For each shipment, determine the number of units of capacity to reserve on your boat in order to carry all of the additional packages for the various factions in the worst case.

Input

The input library is called "shipping"; see the sample inputs below for examples in your language. It defines 6 methods:

- NumberOfVillages()
- VillageFaction(village_index)
- VillageImmediatelyDownstream(village_index)
- NumberOfShipments()
- GetShipmentSource(shipment_id)
- GetShipmentDestination(shipment_id)

Villages and shipments are both zero-indexed. VillageImmediatelyDownstream returns the index of the village that lies immediately downstream, except for the village at the mouth of the river system, whose return value is the index of this village (since no other village is downstream, just the sea). A single call of VillageImmediatelyDownstream will take approximately 0.04 microseconds. A single call of VillageFaction will take approximately 0.02 microseconds.

Output

Output a space-separated list of integers, where the i-th integer is the minimum number of units of capacity needed for the boat delivering the i-th shipment to carry all additional packages in the worst case.

Limits

Each node will have access to 256MB of RAM, and a time limit of 5 seconds.

$1 \leq \text{NumberOfVillages}() \leq 10^8$

$0 \leq \text{VillageFaction}(i) \leq 10^6$ for all villages

$0 \leq \text{GetShipmentSource}(i), \text{GetShipmentDestination}(i) < \text{NumberOfVillages}()$
for all shipments

The `VillageImmediatelyDownstream` method will describe a tree (that is, there will be only one path between any two villages).

Your solution will run on 100 nodes in both inputs.

Small Input

$1 \leq \text{NumberOfShipments}() \leq 10$

$\text{VillageImmediatelyDownstream}(i) \leq i$ for all villages

Large input

$1 \leq \text{NumberOfShipments}() \leq 20,000$

Sample

Input	Output
See below for sample input files.	For sample input 1: 1 1 2 For sample input 2: 1 1 0 For sample input 3: 1 0 0

Sample input libraries:

Sample input for test 1: [shipping.h](#) [CPP] [shipping.java](#) [Java]

Sample input for test 2: [shipping.h](#) [CPP] [shipping.java](#) [Java]

Sample input for test 3: [shipping.h](#) [CPP] [shipping.java](#) [Java]

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform