Submissions

Testrun
0pt Not attempted **0/4 users** correct (0%)

encoded_sum
6pt Not attempted **13/13 users** correct (100%)
11pt Not attempted **12/12 users** correct (100%)

air_show
5pt Not attempted **14/14 users** correct (100%)
20pt Not attempted **1/4 users** correct (25%)

toothpick_sculpture
10pt Not attempted **9/10 users** correct (90%)
15pt Not attempted **0/3 users** correct (0%)

gold
15pt Not attempted **6/10 users** correct (60%)
18pt Not attempted **4/6 users** correct (67%)

Top Scores

bmerry	65
sevenkplus	65
fhlasek	65
mnbvmar	65
eatmore	52
Merkurev	47
ikatanic	37
tozangezan	32
tmt514	32
wafrelka	22

Problem D. toothpick_sculpture

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.

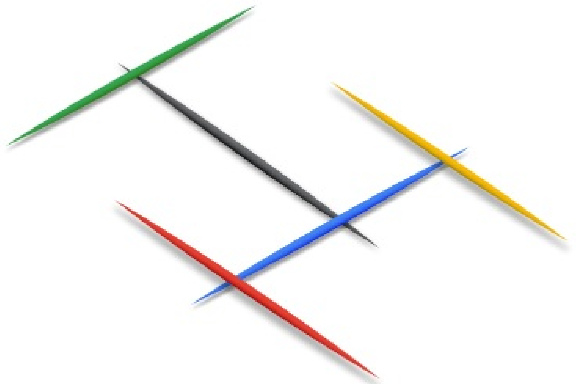
small 10 points *2 minute timeout*	The contest is finished.
large 15 points *10 minute timeout*	The contest is finished.

Problem

Toothpick Sculpture

Your friend Cody-Jamal is an artist. He is working on a new sculpture concept made of toothpicks. Each toothpick has two ends. Each sculpture features a single *foundational toothpick* resting horizontally on the ground. Every other toothpick rests horizontally with its midpoint on one of the ends of exactly one other toothpick. This means each toothpick can have zero, one or two other toothpicks resting on it, forming a binary tree of toothpicks, with the foundational toothpick as the root. Toothpicks do not touch otherwise. Let t_1 be any toothpick, and let t_1 rest on t_2, t_2 rest on t_3, and so on; there is an eventual t_k that is the foundational toothpick.

For instance, in the following picture you can see a sculpture with 5 toothpicks. The black toothpick is the foundational toothpick, resting on the ground. The green and blue toothpicks are resting on the black toothpick, while the red and yellow toothpicks are resting on the blue one.



Cody-Jamal made exactly 1000 sculptures, each using exactly **N** toothpicks, that were exhibited for 1000 days in the largest 1000 cities in the world. As the grand finale, he is planning on placing the foundational toothpicks of 999 of his sculptures on open ends of toothpicks of other sculptures, effectively creating a larger version of a sculpture with the same concept, with 1000 × **N** total toothpicks. He plans on displaying it by placing the foundational toothpick of the structure on top of the Burj Khalifa in Dubai, the tallest building in the world.

Cody-Jamal is enthusiastically telling you his new plan, when your engineering mind notices that the high winds at that altitude will make the structure unstable, so you suggest gluing touching toothpicks together. Cody-Jamal's artistic vision, however, disagrees. The fragility of the construction is part of the conceptual appeal of the piece, he says. You two decide to compromise and use almost invisible carbon nanotube columns to support some toothpicks.

A toothpick (foundational or not) is individually stable if and only if it is supported by a carbon nanotube column. Your math quickly shows that the whole structure can be considered stable if and only if, for each toothpick t that is not the foundational toothpick, either t is individually stable, or t rests on an individually stable toothpick.

Carbon nanotube columns are expensive, though. The cost of the column required to stabilize each toothpick may vary depending on several factors. You decide to write a computer program to assist in choosing a set of toothpicks good enough to make the structure stable while minimizing the sum of the cost of the carbon nanotubes required to stabilize each toothpick.

Input

Toothpicks are numbered with integers between 0 and 1000**N** - 1, inclusive. Integers between 0 and 999 identify the foundational toothpicks of the original sculptures, and 0 is also the foundational toothpick of the large combined sculpture. The numbering of all of the other toothpicks is arbitrary. By way of construction, if you consider a graph made of all toothpicks with two toothpicks being adjacent if they touch each other, the **N** toothpicks corresponding to any original sculpture form a connected subgraph.

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

- **GetNumToothpicks()**:
  - Takes no argument.
  - Returns a 64-bit integer: the number **N** of toothpicks in each original sculpture.
  - Expect each call to take 0.06 microseconds.
- **GetCost(i)**:
  - Takes a 64-bit integer in the range 0 ≤ i < 1000 × GetNumToothpicks().
  - Returns a 64-bit integer: the cost of stabilizing toothpick i.
  - Expect each call to take 0.06 microseconds.
- **GetToothpickAtEnd(i, e)**:
  - Takes two 64-bit integers in the ranges 0 ≤ i < 1000 × GetNumToothpicks(), 0 ≤ e < 2.
  - Returns a 64-bit integer: the id of the toothpick resting on end e of toothpick i in the large sculpture, or -1 if there is no toothpick resting there.
  - Expect each call to take 0.06 microseconds.

Output

Output a single line with a single integer: the minimum sum of the cost to build the necessary carbon nanotubes and make the structure stable.

Recursive solutions, beware! The process stack size is limited to 8 MB and the JVM thread stack size is limited to 1MB. Attempting to change this programatically will result in a Rule Violation.

Limits

Number of nodes: 100. **(Notice that the number of nodes is the same for both the Small and Large datasets.)**
Time limit: 6 seconds.
Memory limit per node: 512 MB.
Maximum number of messages a single node can send: 5000.
Maximum total size of messages a single node can send: 8 MB.
1 ≤ GetNumToothpicks() ≤ 10^6.
1 ≤ GetCost(i) < 10^9, for all i.
-1 ≤ GetToothpickAtEnd(i, e) < GetNumToothpicks() × 1000, for all i, e.
GetToothpickAtEnd(i, e) ≠ 0, for all i, e.
Let *G* be the graph with toothpicks as nodes and two nodes connected if and only if the corresponding toothpicks touch in the sculpture:
G is a connected tree.
Removing the edges between toothpicks 1 through 999 and the toothpicks they are resting on results in exactly 1000 connected components of GetNumToothpicks() nodes each.

Small dataset

GetToothpickAtEnd(i, 1) = -1, for all i.

Large dataset

No additional limits.

Sample

Input	Output

`See input files below.`	`For sample input 1:`
	`250000`
	`For sample input 2:`
	`37500`
	`For sample input 3:`
	`5529586`

Note that the last sample case would not appear in the Small dataset.

Sample input libraries:
Sample input for test 1: toothpick_sculpture.h [CPP] toothpick_sculpture.java [Java]
Sample input for test 2: toothpick_sculpture.h [CPP] toothpick_sculpture.java [Java]
Sample input for test 3: toothpick_sculpture.h [CPP] toothpick_sculpture.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