

Problem D: The Bridges of San Mochti

Source file: `bridges.{c, cpp, java}`

Input file: `bridges.in`

You work at a military training facility in the jungles of San Mochti. One of the training exercises is to cross a series of rope bridges set high in the trees. Every bridge has a maximum capacity, which is the number of people that the bridge can support without breaking. The goal is to cross the bridges as quickly as possible, subject to the following tactical requirements:

One unit at a time!

If two or more people can cross a bridge at the same time (because they do not exceed the capacity), they do so as a unit; they walk as close together as possible, and they all take a step at the same time. It is never acceptable to have two different units on the same bridge at the same time, even if they don't exceed the capacity. Having multiple units on a bridge is not tactically sound, and multiple units can cause oscillations in the rope that slow everyone down. This rule applies even if a unit contains only a single person.

Keep moving!

When a bridge is free, as many people as possible begin to cross it as a unit. Note that this strategy doesn't always lead to an optimal overall crossing time (it may be faster for a group to wait for people behind them to catch up so that more people can cross at once). But it is not tactically sound for a group to wait, because the people they're waiting for might not make it, and then they've not only wasted time but endangered themselves as well.

Periodically the bridges are reconfigured to give the trainees a different challenge. Given a bridge configuration, your job is to calculate the minimum amount of time it would take a group of people to cross all the bridges subject to these requirements.

For example, suppose you have nine people who must cross two bridges: the first has capacity 3 and takes 10 seconds to cross; the second has capacity 4 and takes 60 seconds to cross. The initial state can be represented as (9 0 0), meaning that 9 people are waiting to cross the first bridge, no one is waiting to cross the second bridge, and no one has crossed the last bridge. At 10 seconds the state is (6 3 0). At 20 seconds the state is (3 3 /3:50/ 0), where /3:50/ means that a unit of three people is crossing the second bridge and has 50 seconds left. At 30 seconds the state is (0 6 /3:40/ 0); at 70 seconds it's (0 6 3); at 130 seconds it's (0 2 7); and at 190 seconds it's (0 0 9). Thus the total minimum time is 190 seconds.

Input: The input consists of one or more bridge configurations, followed by a line containing two zeros that signals the end of the input. Each bridge configuration begins with a line containing a negative integer $-B$ and a positive integer P , where B is the number of bridges and P is the total number of people that must cross the bridges. Both B and P will be at most 20. (The reason for putting $-B$ in the input file is to make the first line of a configuration stand out from the remaining lines.) Following are B lines, one for each bridge, listed in order from the first bridge that must be crossed to the last. Each bridge is defined by two positive integers C and T , where C is the capacity of the bridge (the maximum number of people the bridge can hold), and T is the time it takes to cross the bridge (in seconds). C will be at most 5, and T will be at most 100. Only one unit, of size at most C , can cross a bridge at a time; the time required is always T , regardless of the size of the unit (since they all move as one). The end of one bridge is always close to the beginning of the next, so the travel time between bridges is zero.

Output: For each bridge configuration, output one line containing the minimum amount of time it will take (in seconds) for all of the people to cross all of the bridges while meeting both tactical requirements.

Example input:	Example output:
-1 2 5 17 -1 8 3 25 -2 9 3 10 4 60 -3 10 2 10 3 30 2 15 -4 8 1 8 4 30 2 10 1 12 0 0	17 75 190 145 162

Last modified on October 22, 2008 at 10:24 PM.