

	Poor (0.0)	Satisfactory (2.0)	Good (3.0)	Excellent (4.0)
Depth of Discussion (Rx10)	Solution is unclear or non-existent	Solution is slightly explained, but on a very top-level or shallow level of understanding.	Solution/thought process is explained well, but there are aspects that are missing or assumptions that are not explained.	Solution/thought process is explained thoroughly and the logical line of thinking is clear and concise. The line of thought of the author is easy to follow.
Correctness (Rx3)	Solution made no attempt to produce correct input or to run within the appropriate time limit.	Solution is far from a correct solution or is a naive approach that is inefficient. (Note that the term "correct" here is referring to producing the correct output, and not the design principle)	Solution is correct or almost correct, but there are holes or aspects that are questionable or unclear.	Solution is correct and is explained thoroughly and well.
Writing (Rx2)	Writing is full of grammatical and typographical errors. Structure of write up is awkward and disorganized.	There are some grammatical and typographical errors. The writeup has some semblance of structure, but it is confusing to read.	Very few grammatical and typographical errors. Write-up is structured well.	No grammatical and typographical errors. Write-up is exceptionally structured i.e. the flow of discussion is intuitive and smooth..

Problem F: Suspension Bridges

Status: ACCEPTED

Difficulty Rating: 2

Solution Sketch/Write-Up/Narrative:

Summary of Algorithm:

1. Find a low and high point and the middle
2. If greater than middle, make middle new low or else make middle new high
 - a. Get the equality and equal to 0, if positive is greater negative is lower
3. Keep repeating until you find the correct a (s and d are given)
4. Then calculate the sinh and cosh to get the solution

Before we start the discussion, we need to identify the following equations.

$$\begin{aligned}a &= a + s & b &= a * \cosh\left(\frac{d}{2a}\right) \\length &= 2a * \sinh\left(\frac{d}{2a}\right)\end{aligned}$$

In the problem, we have to find the length of a rope that will connect two anchor points with the sagging or the lowest point of the curve of this rope considered and we are only given the distance d between two anchor points and the desired amount s the cable is sagging in the center. To find the other missing variable a , we must use this equality where

$$\begin{aligned}a &= b \\a + s &= a * \cosh\left(\frac{d}{2a}\right)\end{aligned}$$

Now if we equate this equality to 0, where $a - b = 0$, we can place in test values for A to identify if the test A is compatible with the given d and s -- that is the approach we are doing in this problem. First, we set up two pivot points, one is the low point and one is the high point. At the start of the algorithm, we find the low point by getting the lowest possible integer (signed 32-bit) which is -2147483648 and the highest possible integer (signed 32-bit) that is 2147483648. From these two points, we get our first test case, which is the average of these two numbers (the average of two numbers is the middlemost number between them). Given that this test case A and our equality of $a - b = 0$. We can say that if the difference of the equation is greater than 0, then our true A is at the right of the middlemost number A , if it were less than 0, then it would be on the left, if the result is 0 ± 10^{-4} (consideration from the specifications), then this is the A we will use to find the length of the bridge. From this conditional statements, in order to recursively continue the algorithm, if it's greater than 0, make the new low point the previous test A and find the new A , if it's less than 0, make the new high point the previous test A and find the new

A. The algorithm continues until we find the true A to be used to calculate the length. In pseudocode,

```
// int d, s are given

int low = lowest point of int
int high = highest point of int

int findA (low, high)
    int A = (low + high) / 2
    if (a - b > 0)
        findA (A, high)
    else if (a - b < 0)
        findA (low, A)
    else
        return A;
```