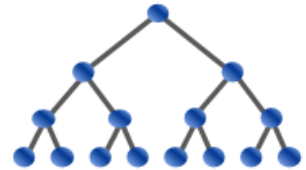


# USA Computing Olympiad

[OVERVIEW](#)[TRAINING](#)[CONTESTS](#)[HISTORY](#)[STAFF](#)[RESOURCES](#)

## USACO 2018 FEBRUARY CONTEST, BRONZE PROBLEM 1. TELEPORTATION

[Return to Problem List](#)

Contest has ended.

### Analysis mode

English (en) ▼

One of the farming chores Farmer John dislikes the most is hauling around lots of cow manure. In order to streamline this process, he comes up with a brilliant invention: the manure teleporter! Instead of hauling manure between two points in a cart behind his tractor, he can use the manure teleporter to instantly transport manure from one location to another.

Farmer John's farm is built along a single long straight road, so any location on his farm can be described simply using its position along this road (effectively a point on the number line). A teleporter is described by two numbers  $x$  and  $y$ , where manure brought to location  $x$  can be instantly transported to location  $y$ , or vice versa.

Farmer John wants to transport manure from location  $a$  to location  $b$ , and he has built a teleporter that might be helpful during this process (of course, he doesn't need to use the teleporter if it doesn't help). Please help him determine the minimum amount of total distance he needs to haul the manure using his tractor.

#### INPUT FORMAT (file teleport.in):

The first and only line of input contains four space-separated integers:  $a$  and  $b$ , describing the start and end locations, followed by  $x$  and  $y$ , describing the teleporter. All positions are integers in the range  $0 \dots 100$ , and they are not necessarily distinct from each other.

#### OUTPUT FORMAT (file teleport.out):

Print a single integer giving the minimum distance Farmer John needs to haul manure in his tractor.

#### SAMPLE INPUT:

```
3 10 8 2
```

#### SAMPLE OUTPUT:

```
3
```

In this example, the best strategy is to haul the manure from position 3 to position 2, teleport it to position 8, then haul it to position 10. The total distance requiring the tractor is therefore  $1 + 2 = 3$ .

Problem credits: Brian Dean

Language:

C ▼

Source File:

 No file chosen

Note: Many issues (e.g., uninitialized variables, out-of-bounds memory access) can cause a program to produce different output when run multiple times; if your program behaves in a manner inconsistent with the official contest results, you should probably look for one of these issues. Timing can also differ slightly from run to run, so it is possible for a program timing out in the official results to occasionally run just under the time limit in analysis mode, and vice versa. Note also that we have recently changed grading servers, and since our new servers run at different speeds from the servers used during older contests, timing results for older contest problems may be slightly off until we manage to re-calibrate everything properly.

