

## RADIO

One way to get from A to B is not to use the memorized frequencies at all. For this we need  $\text{abs}(A-B)$  presses.

Notice that it makes no sense to use more than one memorized frequency; we can immediately jump to the frequency closest to B. For each memorized frequency, we consider jumping to it from frequency A, and then moving to B one by one.

## CRTALO

We process one command at a time. At each moment we need to know the location  $(r, c)$  of the pencil. We also keep track of which squares the pencil went through horizontally and vertically in two separate matrices H and V.

For each command:

- Calculate the new location of the pencil  $(p, q)$
- If  $(p, q)$  is outside the paper, ignore the command – the pencil stays at  $(r, c)$
- If the command is left or right, then set  $H(r, s) = 1$  and  $H(p, q) = 1$
- If the command is up or down, then set  $V(r, s) = 1$  and  $V(p, q) = 1$
- Move the pencil –  $(r, s)$  becomes  $(p, q)$

After processing all commands we need to format the output. For each square  $(i, j)$ :

- Output '.' if  $H(i, j) = 0$  and  $V(i, j) = 0$
- Output '-' if  $H(i, j) = 1$  and  $V(i, j) = 0$
- Output '|' if  $H(i, j) = 0$  and  $V(i, j) = 1$
- Output '+' if  $H(i, j) = 1$  and  $V(i, j) = 1$

The official source codes in C and Pascal show this approach, while the official source code in C++ uses a single two-dimensional matrix for both H and V (using bitwise operations).

## BARD

For each villager we keep track of which songs he or she knows. We will also need to know the number of songs sung by the Bard.

When the bard is present, the song counter is incremented by one and the arrays of all villagers present are updated accordingly. If the bard is not present, then all present villagers learn the union of the songs they know between them.

## BUBA

If the firefly destroys a stalagmite (stalactite) of size X, then it will also destroy all stalagmites (stalactites) of size greater than or equal to X, so the relative ordering of stalagmites (stalactites) is irrelevant. We can therefore separately sort stalactites and stalagmites and use binary search to determine the number of obstacles destroyed on any given level. Iterating on all possible levels gives us the solution.

It is also possible to solve the problem faster, by preprocessing the number of stalagmites (stalactites) of size less than X, for every X.