WOSS Dual Olympiad 2021 S3: Late for Class!

Time limit: 2.0s **Memory limit:** 512M

Having just arrived at WOSS, you are often late to class because you always take the wrong path! Because of this, you have decided to code a program that will help you take the shortest path to all your classes.

WOSS, for our purposes, can be simplified into an $R \times C$ grid. The grid is given to the program with a specific coding system: `.` represents an empty spot you can move through, `#` represents a wall (which you cannot move through) and an uppercase letter represents a portal (because WOSS is definitely technologically advanced enough to have those). Each letter comes in pairs, with one uppercase and one lowercase. If you enter the spot occupied by the capital letter, you will teleport to the spot with the corresponding lowercase letter. So, if you enter a spot labelled as `A`, then you will teleport to the spot marked as `a`. It is guaranteed that all uppercase letters will have a corresponding lowercase letter. Note that teleporters are one-way. Lastly, you cannot pass through a lowercase letter grid space.

For the purposes of this problem, you walk at a constant speed. To take one step in any direction on the grid (up, down, left, right) takes you 1 second. Furthermore, it takes zero time to teleport from one side of a teleporter to the other.

Your current position and classes will also be marked on the $R \times C$ grid. `0` will represent your starting position. The numbers `1`, `2`, `3`, and `4` will mark your first, second, third, and fourth classes, respectively. You must get to each class in that exact numerical order. Note that your starting point after getting to the first class will be the `1`, after getting to the second class, it will be `2`, and so on and so forth.

Your job is, given the grid for WOSS, with period `1`, `2`, `3`, and `4` classes marked, output the smallest possible amount of time you must spend travelling between classes.

Constraints

$1 \le R, C \le 200$

Subtask 1 [10%]

No walls or portals will be present in the grid.

Subtask 2 [30%]

No portals will be present in the grid.

Subtask 3 [60%]

No additional constraints.

Input Specification

The first line of input will contain two integers: R and C. These represent the dimensions of the grid, with R representing the number of rows and C representing the number of columns.

The next R lines of input will consist of C characters. Each character will be either a `.`, `#`, an uppercase letter, a lowercase letter, or an integer from `0`-`4`. Refer to the problem statement for what each of these characters represents.

Output Specification

Output the smallest amount of time (in seconds) required to travel between all `4` classes. The order you must follow is `0` to class `1`, class `1` to class `2`, class `2` to class `3`, and finally class `3` to class `4`. If it is impossible to get to any of the classes, output `-1`.

Sample Input 1

```
7 7
.....3.
.......
.0.....
.......
......1
...2...
.....4.
```

Sample Output 1

```
24
```

Sample Input 2

```
10 10
##########
#0.....1.#
#....###.#
#..3.#2..#
#....###.#
#........#
#..###...#
#..#4....#
#..##....#
##########
```

Sample Output 2

```
31
```

Explanation for Sample Output 2

0 → 1 = 6 steps

1 → 2 = 5 steps

2 → 3 = 11 steps

3 → 4 = 9 steps

Sample Input 3

```
17 18
...#X#.........#x4
####.#.........###
#k1..#............
####.#............
...#.#....j2......
...#J#............
...###............
..................
..................
..................
...........#......
...........#..##..
...........#..#K..
...........#..##..
....3......#.####.
...............0#.
.........########.
```

Sample Output 3

```
66
```