Overview: Design a configuration of towers to maximize the damage to the enemy.

Description: Hiro is on a mission to improve his microbots! Although they are already an impressive invention, he wants them to help curb an impending viral attack in an autonomous self-sustaining manner, and he wants your help to finish the task!

Based on gathered intelligence, in order to stop the spread of the attack, the microbots must self-assemble into at most *k* defense towers placed strategically around the *n* x *m* battlefield. Furthermore, experts have gathered evidence to exactly pinpoint the invasion starting at a cell *s* on the *n* x *m* board, and ending in a different cell *t*. A tower in a cell can attack its 8 adjacent cells. Besides towers, you can also place an arbitrary number of obstacles in some cells.

The **damage of a cell** is defined as the number of towers that can attack this cell. The **damage of a path** from *s* to *t* is the sum of damages of all cells on that path (including cells *s* and *t*). After the microbot towers and obstacle placement are finished according to your proposed defense plan, the viral attack will come out from *s*, and will choose a path to *t* that has the **minimum damage of all possible paths**, defined as the **alien damage of a plan**. You, standing in the junction of life and death of mankind, need to design a plan to maximize **enemy damage**, before all hope is lost. Notice that after placing towers and obstacles you have to make sure there is still one path unblocked by any tower or obstacle from *s* to *t*; otherwise, the viruses will build up and form larger attacks in the future.

Filename: tower.{java, cpp, c, cc, py}

Input: The first line contains *n*, *m*, *k*, as described above.
The next *n* lines describe the *n* x *m* board, each line with *m* characters.
The characters correspond to the following:

`.`: empty cell
`S`: entrance of viruses
`T`: exit for viruses
`#`: cell already occupied by obstacles

Output: Output a single integer, the maximum damage you can achieve by placing no more than *k* towers and an arbitrary number of obstacles.

Assumptions: $1 \le k \le 15$
$2 \le n \le 6$, $2 \le m \le 20$
It is guaranteed that there is a path not blocked by any tower or obstacle from *s* to *t*

Sample Input #1:
```
3 3 1
S.T
...
...
```

Sample Output #1:
```
7
```

Explanation #1
```
S#T
.X.
```

```
... (X denotes the tower, and # is an obstacle we added)
```

Sample Input #2:

```
4 4 2
S.#.
....
....
.##T
```

Sample Output #2:

```
9
```

Explanation #2

```
S.##
X.X#
#...
###T
```