

## Solutions

---

### KVIZ

We generate the three hints according to the guidelines laid out in the problem description.

For the first hint we need to enumerate all letters (call this number  $n_{\text{letters}}$ ) and replace them with dots.

The second hint is generated by revealing the first  $\text{round}(n_{\text{letters}}/3)$  letters.

The final hint requires us to reveal all vowels. If there are no vowels to reveal, reveal the second third of all the letters.

### PISOAR

We can model the urinal using a finite state automaton with  $K+L$  states. A state is described by a boolean `infront`, telling us whether the urinal has registered someone standing in front of it (which means that he had previously been standing there for at least  $K$  seconds), and an integer `time`, how long somebody has been standing in front or how long there hasn't been anybody in front of the urinal. The automaton also needs a set of transition rules as follows:

```
(not registered, t) + '0' => (not registered, 0)
(not registered, t) + '1' => (not registered, t+1)
(not registered, K) => (registered, 0)
(registered, t) + '0' => (registered, t+1)
(registered, t) + '1' => (registered, 0)
(registered, L) => flush and proceed to (not registered, 0)
```

### KOZE

We use a flood-fill algorithm (implemented with either BFS or DFS) to enumerate wolves and goats sharing a region. If the number of wolves in a region is greater than or equal to the number of goats, the wolves own that region and get added to the total.

### NULE

Observe that the number of zeroes on some path equals  $\min(\text{fac2}, \text{fac5})$ , where `fac2` and `fac5` represent the number of occurrences of the number 2 and the number 5 in the integer factorization of the cost of that path, respectively. This allows us to find the solution without calculating the actual cost of any path.

From this we can easily prove that the optimal path has either the smallest number of occurrences of the number 2 in the factorization, or the smallest number of occurrences of the number 5.

We can use dynamic programming to find two different paths, one with the minimum number of twos and one with the minimum number of fives on it. The final solution is then the minimum of two numbers, the number of twos on the first path and the number of fives on the second path.

Time complexity:  $\Theta(N^2)$