(After watching the Soln.)

I tried Solving the problem using dfs dp. But I just saw the case where it will not work.

So we use BFS, minimum number of moves which reach the required state is the winner. Cause that is the minimum.

Start your BFS from the node S, and no need to maintain any visited.

```
BFSNode {
    int x;
    int y;
    int energy;
    int bitmask;
}
```

Well this bitmask is representing each of the 'L' that we are covering

Map(Position, index)

$(x_1, y_1) \to 0$
$(x_2, y_2) \to 1$
$\vdots$
$(x_{10}, y_{10}) \to 9$

Only 10 L's are present.

bitmask

Doubt?

what if all L's cannot be covered?

Is there some condition on which we should stop entertaining entry into the queue.
Cause if we just wait for state to reach and end up in infinite condition.

A visited has to be maintained.

As soon as in BFS

(node.bitmask == $2^{10} - 1$)

that's the answer.

if am visiting the cell with same energy & same bitmask then this will happen.

visiting condition

You can maintain a

{visited[x][y][energy][bitmask] = 0 or 1}