

Паякан в беде

Автор задачи и разработчик: Даниил Голов

Как обычно, первая подгруппа решается полным рекурсивным перебором. Достаточно для текущего состояния перебрать все возможные следующие и запуститься от них. В конце из всех найденных путей требуется выбрать минимальный.

Чтобы проверять, возможно ли определенное действие,

- при движении надо проверить, что символ в новой клетке равен ‘.’;
- при повороте надо проверить, что в каждой новой занимаемой клетке символ равен ‘.’, это можно сделать за время k , просто итерируясь по ним.

Проще всего итерироваться в данном направлении, храня массивы $\mathbf{dx} = [1, 0, -1, 0]$ и $\mathbf{dy} = [0, -1, 0, 1]$, тогда сдвиг в направлении d — это сдвиг из (i, j) в $(i + \mathbf{dy}[d], j + \mathbf{dx}[d])$, тогда как поворот по или против часовой стрелки — это изменение d на ± 1 .

Во второй подгруппе можно заметить, что для $n, m > 1$ даже при одном рифе на поле один из двух путей «вправо, затем вниз» и «вниз, затем вправо», всегда свободен (если риф не в $(1, 1)$ и не в (n, m)), поэтому ответ равен $n + m - 2k + 1$ ($n - k$ и $m - k$ перемещений, и один поворот). Если же на доске есть риф, и $n = 1$ или $m = 1$, либо если риф занимает первую или последнюю клетку, то пути нет, и ответ на задачу — «-1».

Для решения следующих подгрупп стоит воспользоваться поиском в ширину для нахождения минимального пути в графе. Вершинами в графе будут состояния (i, j, d) — позиция и текущее направление. Из каждой вершины ведет не более четырех ребер — движение по направлению и против направления и поворот по или против часовой стрелки. Требовалось просто аккуратно реализовать рассмотрение этих четырех ребер, для каждого из новых состояний проверив, возможно ли оно.

Для решения последней подгруппы нельзя проверять возможность поворота за $\mathcal{O}(k)$. Требуется быстро проверять, свободны ли от рифов определенные k клеток подряд, или нет. Для этого предподсчитаем $\mathbf{right}[i, j]$ и $\mathbf{down}[i, j]$ — максимальное количество свободных подряд клеток, начиная от (i, j) вправо и вниз, соответственно.

Эти значения можно вычислить динамическим программированием: для клеток, занятых рифами, эти значения равны 0, а для остальных $\mathbf{right}[i, j] = \mathbf{right}[i, j+1] + 1$ и $\mathbf{down}[i, j] = \mathbf{down}[i+1, j] + 1$. Используя эти значения, можно быстро проверять возможность поворота. Полное решение имеет асимптотику времени работы $\mathcal{O}(nm)$.