

Тестовое задание

Задача о муравье на координатной сетке

Автор: Яковлев А. И.

Оглавление

Задание.....3

Общее описание проекта.....3

Тестовое приложение.....3

Описание решения.....3

Задание

На бесконечной координатной сетке находится муравей. Муравей может перемещаться на 1 клетку вверх $(x, y+1)$, вниз $(x, y-1)$, влево $(x-1, y)$, вправо $(x+1, y)$, по одной клетке за шаг.

Клетки, в которых сумма цифр в координате X плюс сумма цифр в координате Y больше, чем 25, недоступны муравью. Например, клетка с координатами (59, 79) недоступна, т.к. $5+9+7+9=30$, что больше 25.

Сколько клеток может посетить муравей, если его начальная позиция (1000,1000), (включая начальную клетку)?

Общее описание проекта

Проект рассчитан под сборку через `stake` стандарт `c++ 17`. Сборка осуществлялась в `MSVS 2022`. Исходный код разделен на главный файл `main.cpp` и каталог `src` содержащий класс солвера `AntMovements.cpp/h`. В каталоге `out` исполняемый файл под `windows`.

Тестовое приложение

Тестовое приложение выполнено по классической схеме, функция `main` вызывает два теста:

- `RunDebugTest` – Короткий тест на максимальной сумме 12. Тест выводит состояние матрицы по которому можно судить о правильности работы алгоритма и количество доступных ячеек. В выводе матрицы символ 'O' показывает доступные ячейки, 'X' ячейки недоступных островов, пустота места с максимальной суммой.
- `RunTaskTest` — Полный тест на максимальной суме 25, вводит только количество доступных ячеек.

Описание решения

Решение не рассматривает случаи когда базовая координата не равна 1000,1000, в целом это можно сделать, это несколько изменится расчет «игрового» поля, сам же принцип подсчета доступных ячеек никак не изменится. В конструкторе класса вычисляется размер поля по X и Y через метод `FillinMatrix`. Затем заполняется двумерный массив содержащий ячейки поля. Каждая ячейка содержит структуру `MatrixCell`.

`MatrixCell`:

- `coordX` – координата по X для отладки
- `coordY` – координата по Y для отладки
- `sumValue` – сумма или -1 если сумма \geq предельному значению.
- `visited` – состояние ячейки — была или нет посещена в процессе анализа.

Поле представляет собой серию островов пилообразной формы. Вблизи ордигина острова частично сливаются, дальше разъединяются и уменьшаются в размере. Суть задачи заключается в нахождении площади смежных островов идущих от ордигина (левый верхний

угол на картинке)

[illegible]

Показан случай для максимальной суммы 12.

Метод `CalculateAccessibleFieldSize` производит анализ поля на предмет доступности муравью. Это делается методом распространяющейся волны (похож на обход дерева в ширину). Для исходной точки 0,0 ищутся индексы всех соседей (Соседи берутся в форме +, в центре текущий элемент) и заносятся в список `indicesToVisit`. Затем в цикле из списка извлекается первый элемент с удалением из списка. Для каждого из индексов содержащихся в этом элементе производится извлечение элемента из массива. Затем производится проверка на посещенность элемента и возможность прохода по нему. Если элемент не посещен и проход возможен, то для этого элемента так же ищутся соседи и добавляются в `indicesToVisit`, сам элемент помечается как посещенный и увеличивается счетчик доступных для посещения ячеек `reachableCellCount`. Затем цикл повторяется до тех пор пока `indicesToVisit` не станет пустым. При завершении работы метод возвращает `reachableCellCount`.