

Тестовое задание “Муравей на координатной сетке”

Задание:

На 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)?

Решение выполнено в Code::Blocks 20.03 (Windows x64);
Стандарт C++ 17;

Структура проекта AntPath:

- 1) Headers files :
 - ant_path.h
 - check_move.h
- 2) Source files:
 - check_move.cpp
 - ant_path.cpp
 - main.cpp
- 3) Сmake файл для сборки проекта

Решение поставленной задачи является универсальным и может использоваться с любыми начальными координатами и величины вершины на которую он может попасть.

Файл <ant_path> и <ant_path.cpp> содержат:

- 1) Структуру с координатами X и Y
 - конструктором по умолчанию (по условию задачи <1000, 1000>)
 - и конструктором для установки любых начальных координат
- 2) Сам класс пути муравья **AntPath** и его методы
 - конструктор **AntPath(Coordinates& now)** принимающий структуру с начальными координатами и занесением их в контейнер как начальной позиции

- метод **void ExploreWorld()**, вычисляющий все возможные вершины которые может посетить муравей и запускающий методы движения по всем возможным направлениям **MoveUp()**, **MoveDown()**, **MoveLeft()**, **MoveRight()**
 - метод **void MoveUp()** - движение вверх с проверкой на увеличение координаты Y
 - метод **void MoveDown()** - движение вниз с проверкой на уменьшение координаты Y
 - метод **void MoveLeft()** - движение влево с проверкой на уменьшение координаты X
 - метод **void MoveRight()** - движение вправо с проверкой на увеличение координаты X
 - метод **void Print()** - выводящий координаты всех вершин которые мо
 - получение результата счетчика вершин **int GetCountCells()**
 - основной контейнер содержащий координаты всех вершин, которые может посетить муравей **std::set<Coordinates> all_cells_**
 - счетчик вершин, которые может посетить муравей **int count_cells_**
- 3) Переопределение оператора “<” для контейнера Set , используемого в самом классе как контейнер для добавления структур с координатами шагов муравья
- 4) И их реализации в файле **<ant_path.cpp>**

Файл **<check_move.h>** и **<check_move.cpp>** содержат:

- 1) Функцию “**bool IsPossible(int x, int y)**” для проверки возможности движения муравья на вершину с данными координатами
- 2) Функцию “**int DigitsSum(int x, int y)**” возвращающую сумму чисел координат по условию задачи
- 3) Их реализацию в **<check_move.cpp>**
- 4) Константная переменная **const static int MAX_MOVE_** представляющая собой максимальную сумму чисел в координатах вершин, которая доступна муравью для движения

Описание решения:

В файле **main.cpp**

Создаём экземпляр структуры **Coordinates** с начальными координатами (1000, 1000)

Создаём экземпляр класса **AntPath** с добавлением через конструктор начальных координат муравья и запускаем метод класса **ExploreWorld()** для нахождения всех вершин которые может посетить муравей.

Метод **ExploreWorld()** содержит очередь (std::queue) для занесения координат вершин, которые надо посетить и запускает проверку движения муравья по 4-м направлениям.

Каждая занесенная в очередь вершина также проходит проверку движения по 4-м направления и проверку контейнера **std::set <Coordinates> all_cells_** на то занесена ли уже вершина в список координат которые может посетить муравей. В проекте переопределен оператор “<” для того чтобы контейнер знал как ему упорядочивать элементы в виде структур Coordinates. Проверка вершин, которые добавляются в очередь осуществляется пока это возможно (пока очередь не окажется пуста), для этого используется цикл while.

Также в классе предусмотрен счетчик вершин которые может посетить муравей **count_cells_** который увеличивается каждый раз когда вершина заносится в основной контейнер с вершинами, которые может посетить муравей.

Разделение метода класса **ExploreWorld()** на 4 метода (**MoveUp()**, **MoveDown()**, **MoveLeft()**, **MoveRight()**) представлено для более интуитивного чтения кода.