

Закладка фигур домино

Подготовил студент группы
6371

Кравченко Алексей Николаевич
Факультет КТИ

Поставленные задачи

1. Определить возможность закладки поля фигурами домино.
2. Вычислить количество способов закладки фигур домино.
3. Наполнение базы данных количеством способов закладки.

Используемые программы.

- IDE IntelliJ Idea(java)
- MySQL(SQL Workbench)

Алгоритм работы программы

Считаются два целых числа, a – количество строк, b – количество столбцов.

Далее создается двумерный массив от a и b , где 1 – возможное место нахождения одной половины костяшки домино и 0 – в этой клетке ничего не может находиться.

Возможность закладки поля фигурами домино

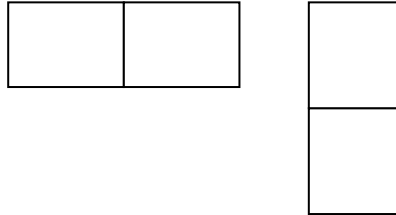
При первом введении данных количество доступных клеток проверяется на четность.

Так же существует теорема о шахматной раскраске, т.е. если если костяшки домино занимают одинаковое количество белых и черных клеток, то такое поле можно заложить костяшками домино

Далее при вызове основной функции(об этом будет рассказано позже) проверяется сколько клеток в итоге остается, если остается одна клетка(без соседних) то на таком поле раскладка невозможна

Количество способов закладки фигур домино

Всего возможно два способа расположения
костяшек домино:



Для вычисления искомого значения вызывается рекурентная функция, в которой реализуется полный перебор комбинаций.

Если не остается двух соседних свободных клеток то функция либо возвращает единицу, т.е. способ учтен, либо 0 и выход из программы, т.е. нельзя наполнить поле фигурками домино.

Наполнение базы данных количеством способов закладки.

Предварительно создается таблица, состоящая из 5 столбцов(id, количество единиц в таблице, количество вершин у фигуры или множества фигур, строка состоящая из координат вершин и количество способов закладки)

Для каждой клетки определяется является ли она вершиной, т.е. вокруг нее три свободные клетки в виде прямого угла.

После этого определяются минимальные координаты x и y для вершин и все поле передвигается к началу координат.

Весь массив вершин переводится в строку.

Наконец идет проверка на наличие такой фигуры в

Основной псевдокод

Аргументы функции:

mas – массив основного поля(из 1 и 0)

a,b – границы таблицы

sum – количество свободных клеток

Функция:

spos(mas, a, b, sum)

Псевдокод:

mas1 = mas;

mas2 = mas;

ЦИКЛ по i<a

 ЦИКЛ по j<b

 bool f=true, f1=true; //переменные для определения

 ЕСЛИ(mas[i][j]=1)ТОГДА{

 mas1[i][j] = 0;

 mas2[i][j] = 0;}

 ЕСЛИ((j<b-1) И (mas2[i][j+1] == 0) ИЛИ (j>=b-1)){f=false;}

 ЕСЛИ((i<a-1) И (mas1[i+1][j]==0) ИЛИ (i>=a-1))ТО{f1=false;}

 ЕСЛИ(f=true)ТО{mas1[i+1][j] = 0;}

 ЕСЛИ(f=true)ТО{mas2[i][j+1] = 0;}

 ЕСЛИ(f И f1) ТО вернуть *sposob*(mas1, sum-2, a, b) + *sposob*(mas2, sum-2, a, b);

 ЕСЛИ(f И !f1) ТО вернуть *sposob*(mas2, sum-2, a, b);

 ЕСЛИ(!f И f1) ТО вернуть *sposob*(mas1, sum-2, a, b);

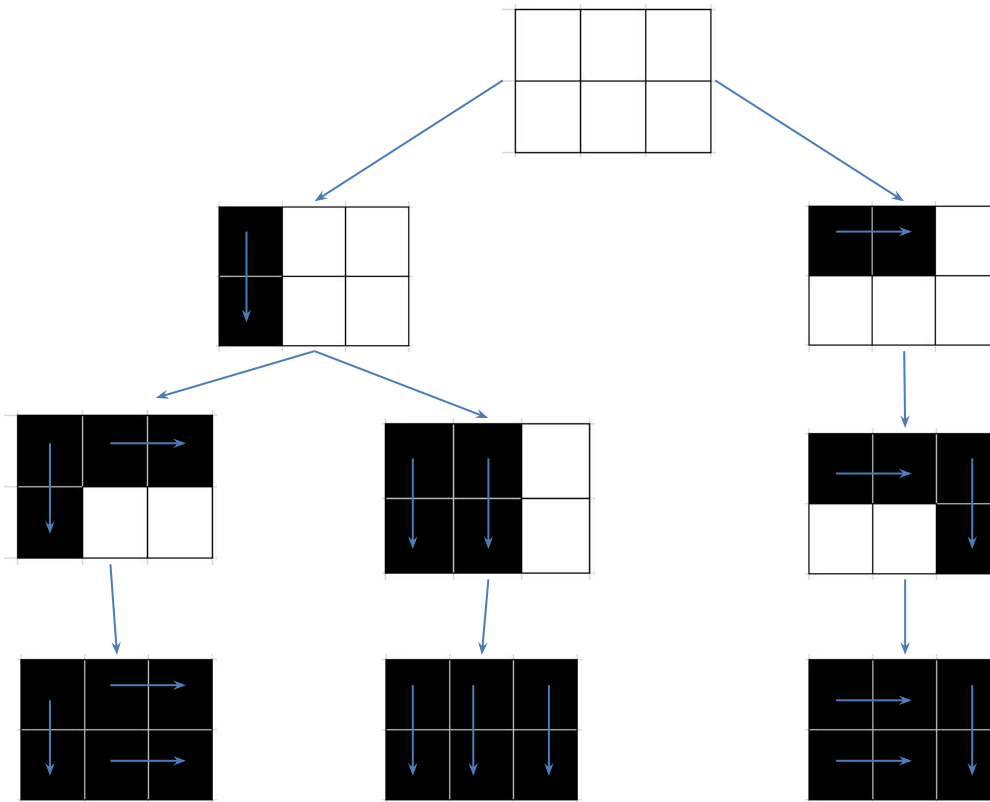
 }

КЦ

КЦ

ЕСЛИ(sum==0)ТО{вернуть 1}ИНАЧЕ{вернуть 0}

Пример №1



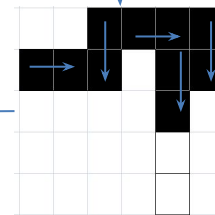
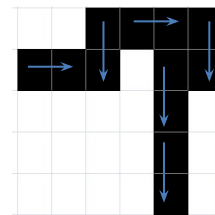
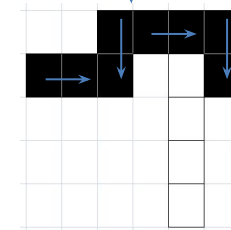
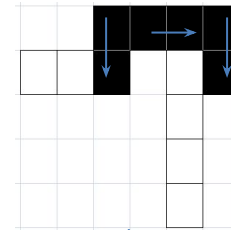
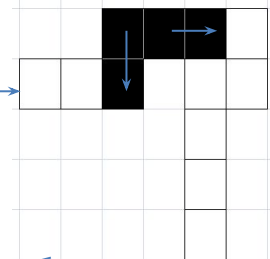
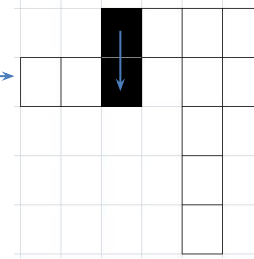
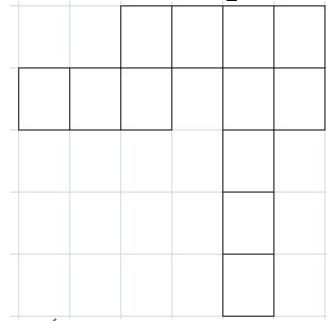
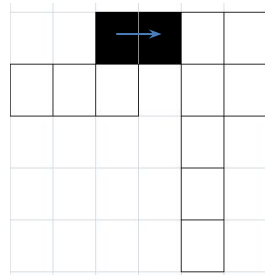
Незакрашенные прямоугольники
с рамками возможное
местоположение
костяшки.
Закрашенные черным уже занятые
клетки.

Из примера видно что программа
работает полным перебором
вариантов
Программа выведет на экран – 3

Пример №2

Прямоугольники с рамками
возможное местоположение
костяшки.
Закрашенные черным уже
занятые
клетки.

Дальнейшая
работа
с этим полем
невозможна
остались три
свободные клетки
Возвращается 0



Возвращается 1(все поля
закрашены)
И т.к. больше вариантов нет,
то программа выводит 1