**АНАЛИЗ НА ЗАДАЧА**

**ИГРАТА 18**

Опорните точки на триъгълната мрежа са 10 на брой и са номерирани от 1 до 10 отгоре надолу и отляво надясно. Свързващите ребра са 18 и се дефинират с начална и крайна точка. Началната точка е винаги с по-малък номер. Масивът int edge[18][2]; съдържа описанието на 18-те ребра със задаване на начална и крайна точка. За да следим покриването на ребрата с клечки, използваме матрицата на съседство line[11][11]. Ако line[i][j] има стойност 0, върху реброто edge[i][j] няма поставена клечка. Ако line[i][j] има стойност 1, върху реброто edge[i][j] има поставена клечка.

Ребрата на мрежата заграждат 9 малки триъгълника, които не се припокриват. Те са номерирани отгоре надолу и отляво надясно с номера от 0 до 8.

Структурата на триъгълната мрежа се описва с масива от структури от тип Triangle. Това е масива triang с 10 елемента (нулевия ред не се използва).

Например:



**typedef struct{**

**int to[3];** // списък на възможните втори точки за ребра с първа точка, съвпадаща с номера на реда. Празните позиции съдържат -1.

**int triang[3][2];**// за всяка втора точка – номерата на триъгълниците, за които реброто е обща страна.

**}Triangle;**

Да разгледаме масива triang от структури от тип Triangle:

**Triangle triang[10] = {**

**{ {-1,-1,-1}, {{-1,-1}, {-1,-1}, {-1,-1}} },**

**/\*From 1 -> to\*/ { { 2, 3,-1}, {{ 0,-1}, { 0,-1}, {-1,-1}} },**

**/\*From 2 -> to\*/ { { 3, 4, 5}, {{ 0, 2}, { 1,-1}, { 1, 2}} },**

**/\*From 3 -> to\*/ { { 5, 6,-1}, {{ 2, 3}, { 3,-1}, {-1,-1}} },**

**/\*From 4 -> to\*/ { { 5, 7, 8}, {{ 1, 5}, { 4,-1}, { 4, 5}} },**

**/\*From 5 -> to\*/ { { 6, 8, 9}, {{ 3, 7}, { 5, 6}, { 6, 7}} },**

**/\*From 6 -> to\*/ { { 9,10,-1}, {{ 7, 8}, { 8,-1}, {-1,-1}} },**

**/\*From 7 -> to\*/ { { 8,-1,-1}, {{ 4,-1}, {-1,-1}, {-1,-1}} },**

**/\*From 8 -> to\*/ { { 9,-1,-1}, {{ 6,-1}, {-1,-1}, {-1,-1}} },**

**/\*From 9 -> to\*/ { {10,-1,-1}, {{ 8,-1}, {-1,-1}, {-1,-1}} }**

**};**

Към примера с начална точка на ребро 5:

**/\*From 5 -> to\*/ { { 6, 8, 9}, {{ 3, 7}, { 5, 6}, { 6, 7}} },**

Индексът 5 сочи към горния ред на масива triang.

Този ред описва ребра (5, 6), (5, 8) и (5, 9).

Ребро (5, 6) е обща страна на триъгълници с номера 3 и 7.

Ребро (5, 8) е обща страна на триъгълници с номера 5 и 6.

Ребро (5, 9) е обща страна на триъгълници с номера 5 и 9.

int filled[9]; Всеки елемент съдържа броя на страните на съответния триъгълник, които са запълнени с поставени кибритени клечки.

Например, ако разглеждаме ребро (3, 6), което принадлежи на масива edge, тестваме line[3][6] – ако е 1, реброто е запълнено с клечка. Ако е 0 – реброто е свободно.

Алгоритъм:

1. Инициализираме статичната част на мрежата.
2. Въвеждаме описанието на предварителните ходове, като въвеждаме входните данни. С други думи поставяме клечките от частично започнатата игра по местата им.
3. Итерираме по свободните ребра и следим промените в масива filed.
4. Ако някой от елементите на масива filed достигне стойност 3, отброяваме триъгълник в брояча res, включваме реброто в модела (в масива line поставяме 1 на съответното място) и правим следваща итерация.
5. Ако за дадена итерация не е попълнен триъгълник прекъсваме цикъла и отпечатваме res.

София Автор:П.Панов

17.04.22