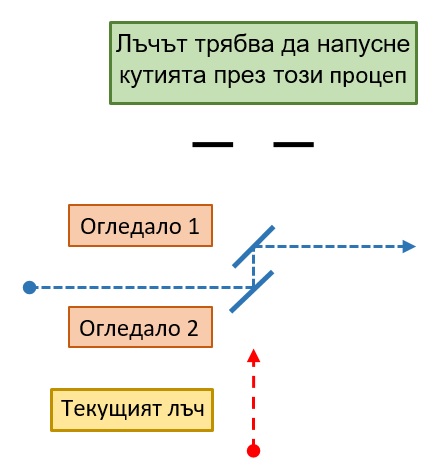
### Анализ на решението на задача

МАГИЧЕСКА КУТИЯ

За решаване на задачата прилагаме лаком (greedy) алгоритъм. Идеята е да трасираме пътя на светлинния лъч, като поставяме огледала по такъв начин, че светлинният лъч да минава колкото е възможно по-нагоре. Ще трасираме последователно лъчите, пуснати в отвори с номера 1, 2, ... , N+M.

За трасиране на пътя на всеки лъч прилагаме следния алгоритъм:

1. Нека текущата позиция на следения лъч е (x - колона и y - ред на клетката). Нека x', y' е колоната и реда на клетката, която съдържа отвора, през който лъчът трябва да напусне кутията.
2. Ако (x == x ' && y == y') == true, тогава изход.
3. Ако лъчът е насочен вьв вертикална посока, отиди на точка 5.
4. Ако има огледало над текущото положение и x ≠ x ' или y = y', отиди една клетка вдясно: x =x+1. В противен случай постави огледало в клетката с координати (х, y) и отиди една клетка нагоре: y=y-1. Отиди на точка 2.
5. Ако има огледало в текущата клетка, отиди надясно: x =x+1. В противен случай нагоре: y=y-1. Отиди на точка 2.

Единствената конфликтна ситуация възниква в следния случай: Текущият лъч трябва да напусне кутията през горния отвор, но по пътя си среща огледало 2. То променя посоката на лъча от вертикална в хоризонтална и по този начин отворът ще бъде пропуснат. Такава ситуация не може да се случи, защото от горната страна на огледало 1 не се отразява светлина, а би трябвало. Според нашия алгоритъм огледалото се поставя само когато то отразява светлинен лъч. Следователно Огледало 1 вече е изпратило лъча, който е наложил неговото поставяне към отвора.

**Реализация:**

Входните данни се съхраняват в променливите sizex, sizey (размери на кутията) и едномерен масив inout (съответствието: вход/изход за всеки пореден отвор).

Масивът bool mirrorInCol се използва за маркиране на наличието на огледало в съответната колона: (true – в колоната има огледало; false – колоната е празна).

**Функцииите:** int numToX(int nr) и int numToY(int nr) пресмятат (x,y) на отвор с номер nr.

**Функцията:**

bool DirOfHole(int nr) връща true – ако през отворът nr може да се пусне хоризонтален лъч (т.е. отворът се намира на вертикална околна стена) и false в обратния случай. Тази функция задава стойност на работната променлива horiz, която играе съществена роля при реализацията на алгоритъма.

**Функцииите:**

void traceH(int, int, int, int, bool) и void traceV(int, int, int, int, bool) осъществяват постъпково трасировката, като се извикват една – друга рекурсивно при промяна на посоката. Рекурсията се прекъсва, когато върхът на лъча достигне изходния отвор.

**Функцията:**

void findMirrors() върти цикъл по отворите на лявата вертикална стена, стартирайки функцията traceH() и след това цикъл по отворите на долната хоризонтална стена, стартирайки функцията traceV().

Автор: Пано Панов