**Пример 1.**

**Из листа клетчатой бумаги размером N x M удалили некоторые клетки размером 1 х 1. Определите, на сколько связных кусков распадается оставшаяся часть листа. К - количество удалённых клеток, (х1; у1), …, (хk; yk) - координаты удалённых клеток.**

**Для решения данной задачи рассмотрим двумерный массив N x M , где удалённые клетки имеют значение -1, а неудалённые - 0.**

**В нашем алгоритме мы будем находить первую клетку со значением 0, помечаем её определённым образом, а потом находим и помечаем клетки, имеющие общую сторону с найденной клеткой и к тому же со значением 0. Потом находим и помечаем «нулевых» соседей последних найденных клеток и т.д. Процесс заканчивается, когда никаких новых клеток не удаётся найти. Значит, мы получили единый кусок листа.**

**Далее выбираем новую начальную клетку и опять повторяем процесс нахождения её «соседей», равных 0.**

**Алгоритм закончится, когда на «листе» не останется клеток со значением 0.**

**Для решения задачи будем использовать очередь. В очереди будем хранить вновь помеченные клетки.**

**Для примера рассмотрим лист размером 4 х 5, где удалёнными (со значением -1) считаются клетки с координатами (1;5), (4;5), (3;4), (2;3), (2;2), (1;1), (3;1). Итак,**

|  |
| --- |
|  |
| i \ j | 1 | 2 | 3 | 4 |  |
| 1 | -1 | 0 | -1 | 0 |  |
| 2 | 0 | -1 | 0 | 0 |  |
| 3 | 0 | -1 | 0 | 0 |  |
| 4 | 0 | 0 | -1 | 0 |  |
| 5 | -1 | 0 | 0 | -1 |  |
|  |  |  |  |  |  |

**Теперь найдём первую клетку со значением 0 и занесём её координаты в очередь. При этом в матрицу на соответствующее место ставится номер куска, который рассматривается (в данном случае 1). Первой найдена клетка с координатами (1;2).**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Индексы | | | | 1 | 2 | | 3 | | 4 | | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |  |
| i | | | | 1 |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |
| j | | | | 2 |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |
|  | | | |  |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |
|  |
| i \ j | 1 | 2 | 3 | | | 4 | |  | |
| 1 | -1 | 1 | -1 | | | 0 | |  | |
| 2 | 0 | -1 | 0 | | | 0 | |  | |
| 3 | 0 | -1 | 0 | | | 0 | |  | |
| 4 | 0 | 0 | -1 | | | 0 | |  | |
| 5 | -1 | 0 | 0 | | | -1 | |  | |
|  |  |  |  | | |  | |  | |

**Рядом с этой клеткой нет других, со значением 0. Удаляем эту клетку из очереди. Потом ищем следующую начальную клетку. Это будет клетка с координатами (1;4). Её в таблице помечаем номером 2 (второй кусок). В очередь вносим её координаты и координаты её нулевых «соседей»-клеток, а потом их «соседей», и т.д. Получим следующее:**

|  |
| --- |
|  |
| Индексы | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |  |
| i | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 4 |  |  |  |  |  |  |  |
| j | 2 | 4 | 4 | 3 | 4 | 3 | 4 | 4 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**На каждом шаге после проверки элемента на наличие нужных «соседей» и его пометке в массиве, удаляем его из очереди. Продолжаем процесс до тех пор, пока очередь не станет пуста.**

**Массив-лист примет следующий вид:**

|  |
| --- |
|  |
| i \ j | 1 | 2 | 3 | 4 |  |
| 1 | -1 | 1 | -1 | 2 |  |
| 2 | 0 | -1 | 2 | 2 |  |
| 3 | 0 | -1 | 2 | 2 |  |
| 4 | 0 | 0 | -1 | 2 |  |
| 5 | -1 | 0 | 0 | -1 |  |
|  |  |  |  |  |  |

**Следующий начальный нулевой элемент с координатами (2;1) будет уже помечаться номером 3 (третий кусок). Вносим его координаты в очередь, и действуем по знакомому алгоритму.**

|  |
| --- |
|  |
| Индексы | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |  |
| i | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 2 | 2 | 3 | 4 | 4 | 5 | 5 |  |  |
| j | 2 | 4 | 4 | 3 | 4 | 3 | 4 | 4 | 3 | 1 | 1 | 1 | 2 | 2 | 3 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Массив примет вид:**

|  |
| --- |
|  |
| i \ j | 1 | 2 | 3 | 4 |  |
| 1 | -1 | 1 | -1 | 2 |  |
| 2 | 3 | -1 | 2 | 2 |  |
| 3 | 3 | -1 | 2 | 2 |  |
| 4 | 3 | 3 | -1 | 2 |  |
| 5 | -1 | 3 | 3 | -1 |  |
|  |  |  |  |  |  |

**Больше нулевых элементов нет. Поэтому алгоритм будет завершен. Количество кусков равно 3.**