Mosaic

Time Limit 3000 ms

색종이를 이용해서 모자이크 패턴을 만들려고 한다.

캔버스에 색종이를 무작위로 붙이고 임의의 영역을 선택해서 패턴을 만들 계획이다.

색종이는 height, width가 각각 1~50의 크기를 가지며 0~255의 색깔을 가진다.

캔버스의 크기는 10~1,000 이며 색깔은 0값을 가진다.

create()를 통해서 색종이가 생성 순서대로 캔버스에 놓이게 되며 다른 색종이와 겹쳐질 수 있다.

색종이가 캔버스에 놓일 때 색종이의 id, 좌상단 좌표(y, x)와 height, width 그리고 color가 주어진다.

이미 캔버스에 놓여진 색종이 아래에 놓여지지 않는다.

겹쳐지는 색종이는 투명하지 않아 아래쪽 색종이의 색이 보여지지 않는다. (색깔 0도 마찬가지)

전달되는 id는 0부터 순차적으로 증가하며, 동일한 id로 create()가 호출되지 않는다.

id는 0~9,999값을 가진다.

remove()는 특정 색종이를 캔버스에서 제거한다.

이때 전달 되는 id는 create()에서 생성된 색종이의 id이며

생성되지 않았거나 이미 제거된 색종이의 id로 remove()가 호출되지 않는다.

삭제되는 색종이는 전체 색종이의 20%를 넘지 않는다.

즉 create()를 통해서 생성된 색종이 수가 100이라면 삭제 되는 색종이 수는 20 이하이다.

move()는 특정 색종이의 위치를 옮긴다.

이때 선택된 색종이가 다른 색종이 아래에 있었다 하더라도

옮겨지고 나면 반드시 색종이들 맨위에 놓여야 한다.

즉 다른 색종이 아래에 놓여지지 않는다.

getPattern()을 통해서 주어진 좌표를 좌상으로 하는 5\*5 범위의 패턴을 반환해야 한다.

이 함수는 create(), remove() 그리고 move() 호출 이후에 반드시 불려지며 단독으로 호출될 수도 있다.

초기에 불려지는 create() 호출 횟수를 제외하고

create(), remove(), move() 그리고 getPattern() 함수의 최대 호출 횟수는 20,000회를 넘지 않는다.

test case 1, 2에서는 getPattern() 함수 호출 횟수가 전체 호출 횟수에 포함되지 않는다.

test case 1에서는 create()와 getPattern() 함수만 호출되며

test case 2부터 remove()가 추가되고 test case 4부터 move()가 포함된다.

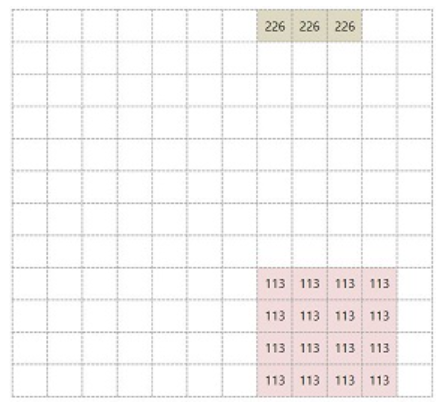
test case 1, 2는 반환되는 패턴을 비교할 수 있도록 입력이 작성되었으며

test case 10까지는 반환된 패턴마다 hash값을 비교하도록 되어 있다.

모든 test case는 각 test case별 hash 조합값을 비교하여 성공/실패를 확인한다.

첫번째 test case를 살펴보면 다음과 같다.

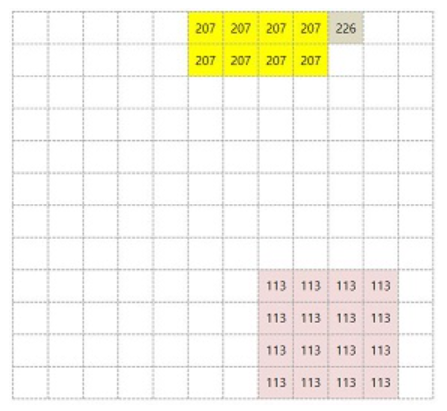
처음 두번의 create()에 의해서 다음과 같은 형태가 된다.



<그림1>

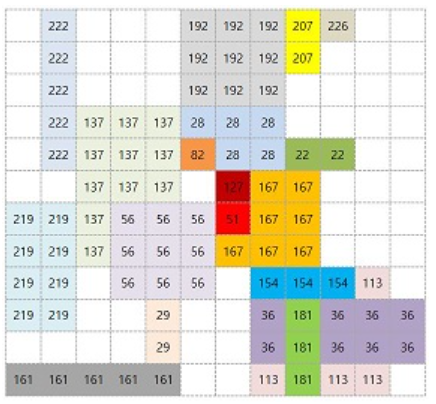
세번째 create()에 의해서 두번째 색종이를 일부 덮게 된다.

노란색으로 되어 있는 부분이 세번째 색종이이며 226이었던 부분 중 두 부분이 덮어 씌워진 상태이다.



<그림2>

이후 총 20번의 create()에 의해서 아래와 같이 된다.

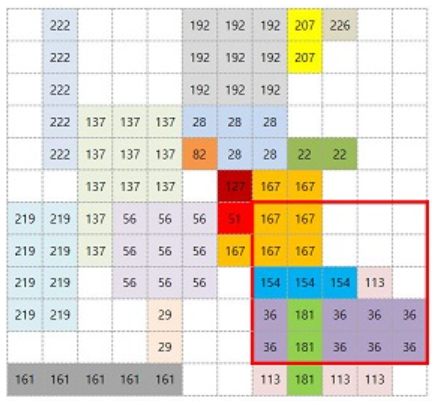


<그림3>

첫번째 getPattern() 함수는 y=6, x=7인 부분을 요청하게 되는데 다음과 같이 빨간 사각형으로 표현된 부분이다.

user code에서 아래 부분을 pattern[5][5]에 복사해서 반환해 주면 된다.

색종이가 없는 부분은 0으로 표현된다. (이 예제 외에 0색을 가진 색종이가 있을 수도 있다.)



<그림4>

초기 create()외에도 command들이 진행되는 동안 create()가 추가 호출 되는데

다음과 같이 수정이 될 수 있다.

빨간색(95)이 새로 쓰여진 부분이며 36, 181 두 부분이 덮어 씌워진 상태이다.

그리하여 <그림4>와 동일한 부분을 getPattern() 했지만 다른 결과를 얻게 된다.

이렇게 동일한 좌표를 요청해도 다른 값이 반환될 수 있다.



<그림5>

test case 1의 최종 형태는 아래와 같다.



<그림6>