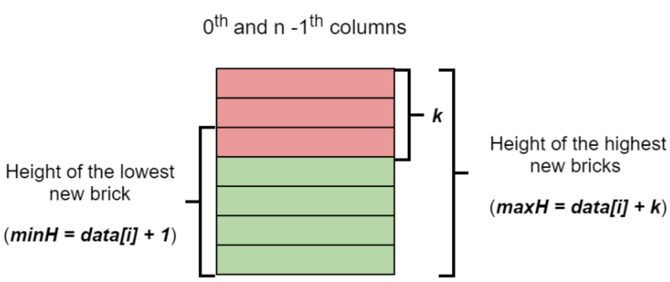
Editorial - Gadol

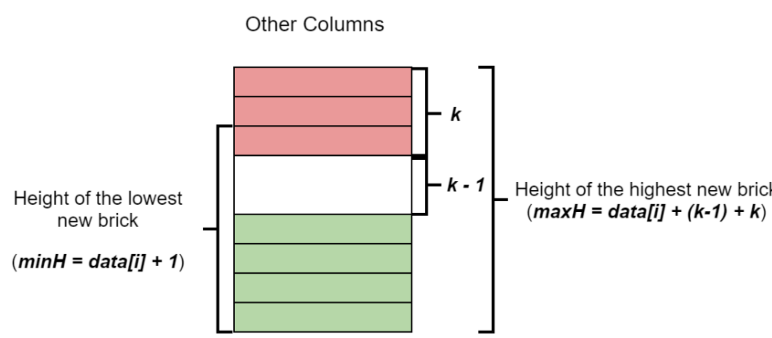
Facts to consider,

1. Every column should contain at least 1 new brick (rule 2)
2. The first and last columns should not have spaces between new and old bricks and can have a maximum of ***k*** new bricks
3. Every other column can have a maximum of ***k-1*** contiguous spaces and a maximum of ***k*** new bricks

By taking each column separately,

- All the columns,
  - Height of the lowest new brick from ground = old height from ground + 1
- 0^{th} and $n-1^{th}$ columns,
  - Height of the highest new brick = old height + number of new bricks
    = old height + ***k***
- Other columns,
  - Height of the highest new brick = old height + no. of spaces + no. of new bricks
    = old height + (***k - 1***) + ***k***





Above heights may violate rule 2 or 3. Therefore, the following steps should be followed to check that.

1. The lowest and the highest height of the new bricks of the 0^{th} column are stored. (rule 3)

2. 1^{st} to $n-2^{th}$ columns are taken one by one and their possible heights are checked with the previous column's heights of the highest and lowest new brick.

   a. If (height of the old bricks of i^{th} column is greater than or equal to the height of the lowest new brick of previous ($i-1^{th}$) column) or (height of the highest new brick of i^{th} column is lesser than the height of the lowest new brick of the previous ($i - 1^{th}$) column) then the second rule is broken. It implies that there is no solution.

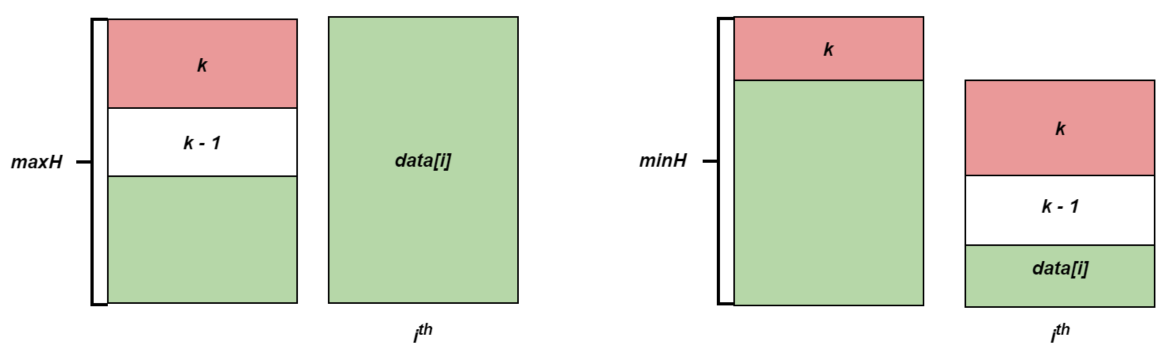
```
if (data[i] >= maxH || data[i] + k + (k - 1) < minH)
       //no solution
```

b. Then height of the highest new brick of i^{th} column is updated as follows,

```
maxH = min(data[i] + k + (k - 1), maxH + (k - 1));
```

c. Height of the lowest new brick of i^{th} column is updated as follows,

```
minH = max(data[i] + 1, minH - (k - 1));
```



3. The last column's ($n - 1^{th}$) heights of highest and lowest bricks are checked.
   a. If (height of the old bricks of last ($n - 1^{th}$) column is already greater than or equal to the height of the lowest new brick of previous ($n - 2^{th}$) column) or (height of the highest new brick of last ($n - 1^{th}$) column is lesser than the height of the lowest new brick of the previous ($n - 2^{th}$) column) then the second rule is broken. It implies that there is no solution.

```
if (data[n - 1] >= maxH || data[n - 1] + k < minH)
     //no solution
```

4. If all of the above cases are false, then it is possible to lay bricks by following all the rules.