1649 . Crie uma matriz classificada por meio de instruções

Hint

Duro 622 75

Empresas

Dado um array inteiro `instructions`, você será solicitado a criar um array classificado a partir dos elementos em `instructions`. Você começa com um contêiner vazio `nums`. Para cada elemento da **esquerda para a direita** em `instructions`, insira-o em `nums`. O **custo** de cada inserção é o **mínimo** dos seguintes:

- O número de elementos atualmente contidos `nums` é **estritamente menor que** `instructions[i]`.
- O número de elementos atualmente nele `nums` é **estritamente maior que** `instructions[i]`.

Por exemplo, se inserir o elemento `3` em `nums = [1,2,3,5]`, o **custo** de inserção é `min(2, 1)` (os elementos `1` e `2` são menores que `3`, o elemento `5` é maior que `3`) e `nums` se tornará `[1,2,3,3,5]`.

Retorne *o **custo total** para inserir todos os elementos* `instructions` *em* `nums`. Como a resposta pode ser grande, retorne-a **módulo** $10^9 + 7$

Exemplo 1:

```
Entrada: instruções = [1,5,6,2]
 Saída: 1
 Explicação: Comece com nums = [].
Insira 1 com custo min(0, 0) = 0, agora nums = [1].
Insira 5 com custo min(1, 0) = 0, agora nums = [1,5].
Insira 6 com custo min(2, 0) = 0, agora nums = [1,5,6].
Insira 2 com custo min(1, 2) = 1, agora nums = [1,2,5,6].
O custo total é 0 + 0 + 0 + 1 = 1.
```

Exemplo 2:

```
Entrada: instruções = [1,2,3,6,5,4]
 Saída: 3
 Explicação: Comece com nums = [].
Insira 1 com custo min(0, 0) = 0, agora nums = [1].
Insira 2 com custo min(1, 0) = 0, agora nums = [1,2].
Insira 3 com custo min(2, 0) = 0, agora nums = [1,2,3].
Insira 6 com custo min(3, 0) = 0, agora nums = [1,2,3,6].
Insira 5 com custo min(3, 1) = 1, agora nums = [1,2,3,5,6].
Insira 4 com custo min(3, 2) = 2, agora nums = [1,2,3,4,5,6].
O custo total é 0 + 0 + 0 + 0 + 1 + 2 = 3.
```

Exemplo 3:

```
Entrada: instruções = [1,3,3,3,2,4,2,1,2]
 Saída: 4
 Explicação: Comece com nums = [].
Insira 1 com custo min(0, 0) = 0, agora nums = [1].
Insira 3 com custo min(1, 0) = 0, agora nums = [1,3].
Insira 3 com custo min(1, 0) = 0, agora nums = [1,3,3].
Insira 3 com custo min(1, 0) = 0, agora nums = [1,3,3,3].
Insira 2 com custo min(1, 3) = 1, agora nums = [1,2,3,3,3].
Insira 4 com custo min(5, 0) = 0, agora nums = [1,2,3,3,3,4].
Insira 2 com custo min(1, 4) = 1, agora nums = [1,2,2,3,3,3,4].
Insira 1 com custo min(0, 6) = 0, agora nums = [1,1,2,2,3,3,3,4].
Insira 2 com custo min(2, 4) = 2, agora nums = [1,1,2,2,2,3,3,3,4].
O custo total é 0 + 0 + 0 + 0 + 1 + 0 + 1 + 0 + 2 = 4.
```

Restrições:

- $1 <= instructions.length <= 10^5$
- $1 <= instructions[i] <= 10^5$