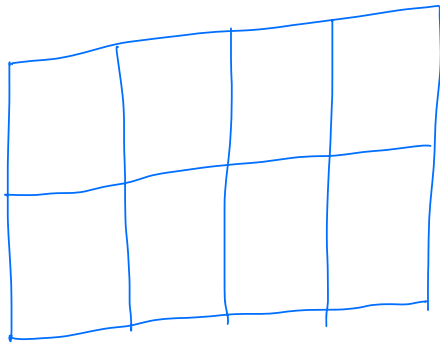


def init-energy-function



I divide the matrix in this way  
Each part will be submitted to a thread

→ gray scale matrix

After calculating, the returning value is the same order with the input.

def update-energy-function

when it removes a column:

it ~~it~~ removes the upper left corner

before removing — the energy function:

← ① 1 2 3 4 5 6

(i,j)  
after removing:

1 2 3 4 5 6

(i,j)

Just (i,j) is influenced, calculate the energy for the pixel

if it removes the first row, except the left corner;

before removing: 0 1 2 ③ 4 5 6 7

after removing: 0 1 ② ④ 5 6 7 8

5's neighbors are still 4 and 6, not influence

2 and 4's neighbors changed

So we calculate 2 and 4 energy again.

**Notice:** We just consider the current line and last line influenced pixel. Because the next line will be calculated later

if the removed pixel is in the first two columns:

before:  $\leftarrow$   $i-1, j$  ① 1 2 3 4 5 6 7

last removed 8 ⑨ 10 11 12 13 14 15

after:  $i, j$   $\rightarrow$  current removing

influenced range

How to judge a pixel is influenced?

See its neighbors change or not!

$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$

they are 5's neighbors

if the removed pixel <sup>(i,j)</sup> is in the last column:

before: 

1	2	3	4	5	6
7	8	9	10	11	12

  
 $i,j$

after: 

1	2	3	4	6
7	8	9	10	11

 influenced

---

if the removed pixel is not near the boundary:

before: 

1	2	3	4	5	6	7
8	9	10	11	12	13	14

  
 $i,j$

case 1:

after: 

1	2	4	5	6	7
8	9	10	12	13	14

  
 $i,j$

influenced range

z's influence is calculated in the last iteration

So we don't calculate z's energy

before: 

1	2	3	4	5	6	7
8	9	10	11	12	13	14

  
 $i,j$

case 2:

after: 

1	2	4	5	6	7
8	9	11	12	13	14

case 3:

before: 1 2 3 4 5 6 7  
 8 9 10 11 12 13 14

after: 1 2 3 4 6 7  
 8 9 10 12 13 14

Diagram showing the removal of column 5 (index 4) from a 2x7 matrix. The 'before' state shows a 2x7 grid with indices 1-14. The 'after' state shows a 2x6 grid where the 5th column has been removed. Brackets and arrows indicate the shift of elements to the right of the removed column.

So, in generally, there are 6 pixels may be influence  
 They are:  $(i, j-2), (i, j-1), (i, j), (i, j+1), (i+1, j-1), (i+1, j)$

The cases for removing row is similar with removing column, just consider the boundary conditions and normal conditions.

def search\_column\_seam:

image pixel matrix:

4	3	7	2	9	1
5	6	9	8	15	2
7	3	2	4	1	14
15	16	2	3	1	7
9	6	5	3	2	4

Diagram showing the removal of column 5 (index 4) from a 5x6 matrix. A red line traces the path of the removed column, and the elements to its right are shifted one column to the left.

dp matrix:

4	3	7	2	9	1
12	6	11	8	16	3
15	11	11	14	4	17
26	27	13	7	15	11
35	19	12	8	7	9

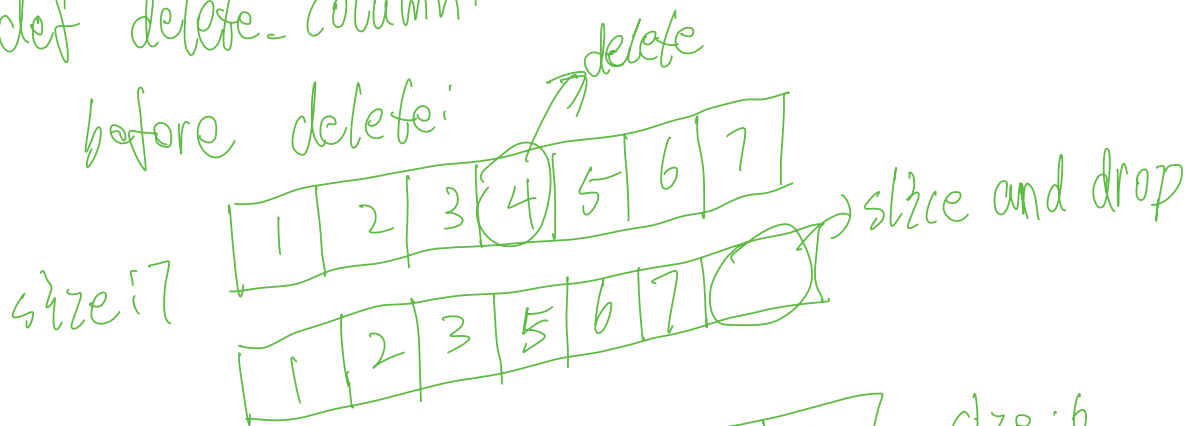
Diagram showing the DP matrix for the column seam removal. The matrix is 5x6. A red line traces the path of the removed column. The values in the matrix represent the cumulative cost of removing a column up to that point. The minimum value is 7, located at the bottom right.

This is the removed column

def search\_row\_seam is similar.

def delete\_column:

before delete:

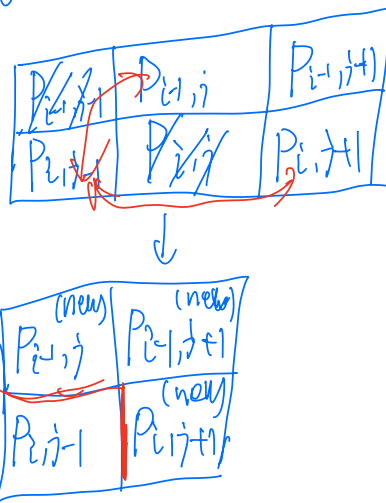


final array: [1, 2, 3, 5, 6, 7]

delete-row is similar

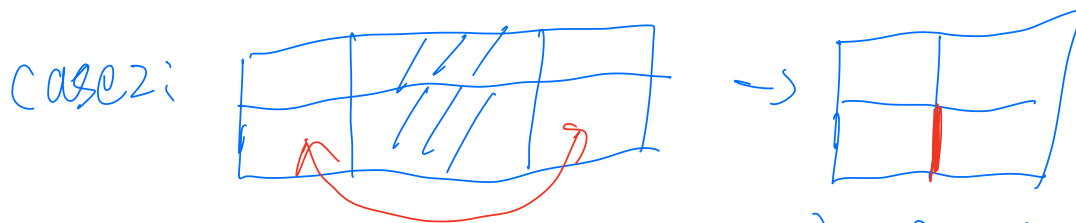
def calculate\_column

Case 1:

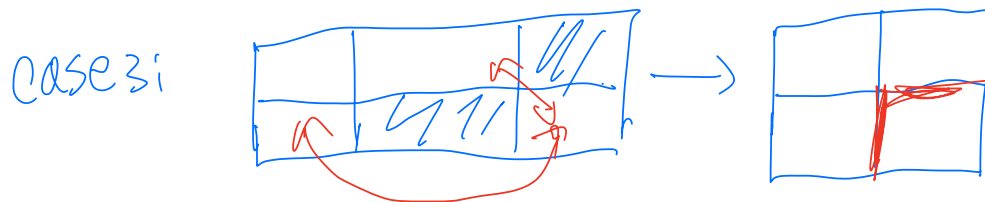


The red lines mean they becomes the new neighbors.

$$\text{Insert energy} = \text{abs}(P_{i,j-1} - P_{i,j}) + \text{abs}(P_{i,j-1}, P_{i,j+1})$$



Insert energy  $= \text{abs}(P_{i,j+1} - P_{i,j+1})$   $P$  is the grayscale



Insert energy  $= \text{abs}(P_{i,j+1} - P_{i,j+1}) + \text{abs}(P_{i,j+1} - P_{i+1,j+1})$

def calculate\_row is similar.