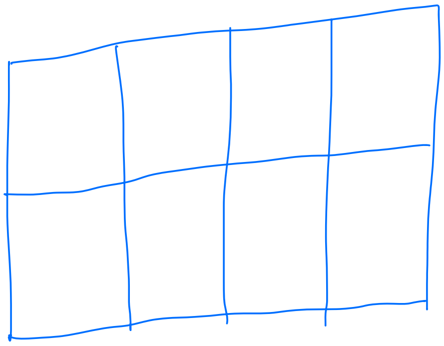
def init_energy_function



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:

if it removes the upperleft corner

before removing — the energy funtion:

← 0 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

i,j

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

---

(i,j)

if the removed pixel is in the first two columns:

before: i-1,j-1 0 1 2 3 4 5 6 7

last removed 8 9 10 11 12 13 14 15

i,j → current removing

after: 1 2 3 4 5 6 7

8 10 11 12 13 14 15

i,j influenced range

How to judge a pixel is influenced?

See its neighbors change or not!

Ex: 1 2 3
4 5 6
7 8 9
they are 5's neighbors

if the removed pixel (i,j) 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 influenced
7 8 9 10 11

---

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

case1:

after: 1 2 4 5 6 7
8 9 10 12 13 14
i,j

influenced range

2's influence is calculated in the last iteration

so we don't calculate 2's energy

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

case2:

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:

		i-1, j+1	i-1, j		
1	2	3	4	6	7
8	9	10	12	13	14
i, j-2	i, j-1	i, j	i, j+1		

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	18
15	16	2	3	1	7
9	6	5	3	2	4

dp matrix:

	4	3	7	2	9	1
5+3	8	6+3 9	9+2=11	8+2=10	16	3
	15	11	11	14	4	17
	26	27	13	7	5	11
	35	19	12	8	7	9
					min	

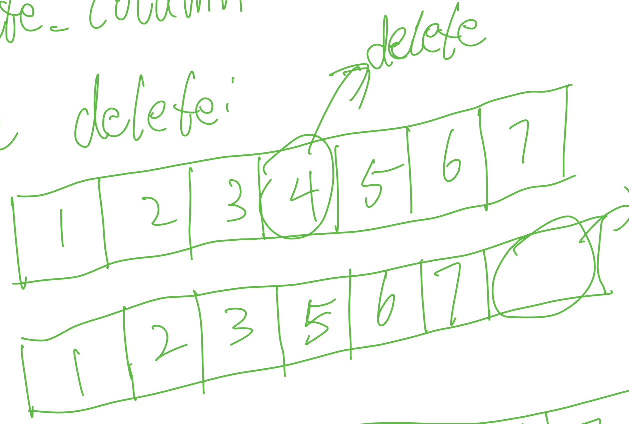
This is the removed column

def search_row_seam is similar.

def delete_column:

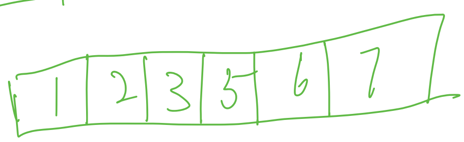
before delete:

size: 7



slice and drop

final array:

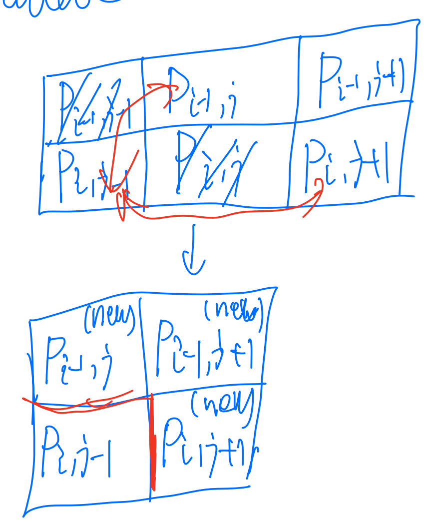


size: 6

delete_row is similar

def calculate_column

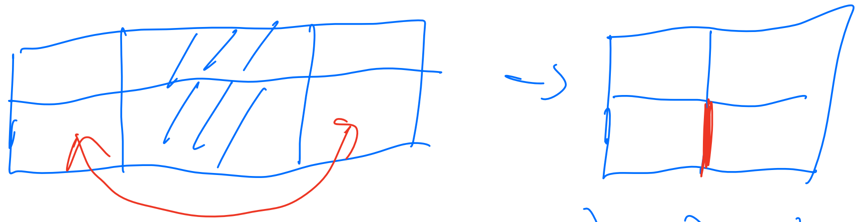
Case 1:



The red lines mean they becomes the new neighbors.

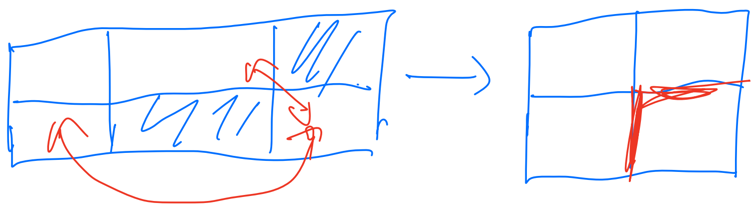
Insert energy = abs $(P_{i,j-1} - P_{i-1,j})$ + abs $(P_{i,j-1}, P_{i,j+1})$

case2:



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

case3:



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

def calculate_row is similar.