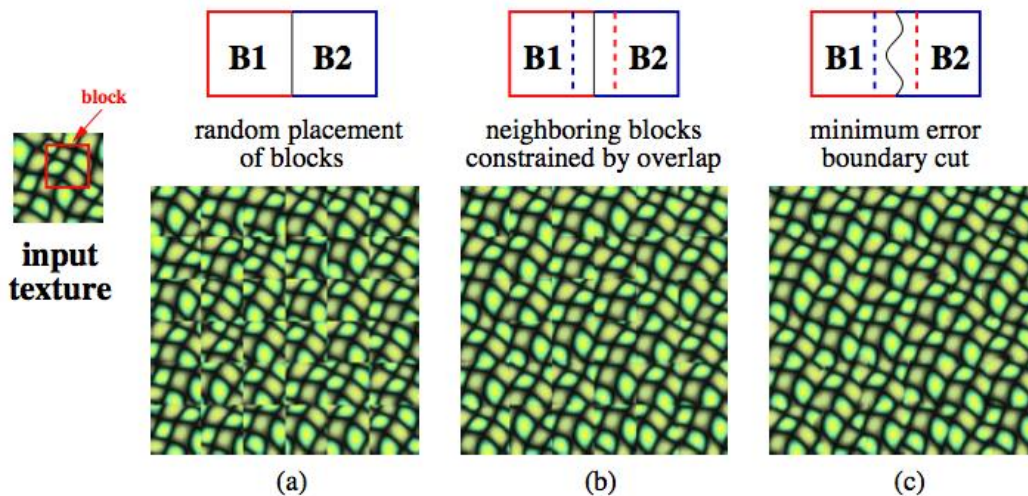


## 문제해결프로젝트 프로젝트 II

### <첫 번째 문제 제시> 이미지 퀼팅 (5점)

이미지 퀼팅(Image Quilting)이란, 하나의 패턴 이미지를 여러 개 이어 붙여서 큰 이미지를 만들어 내는 것을 말한다. 하지만 하위 그림 (a)와 (b)와 같이, 단순히 이미지를 나란히 이어 붙이는 것만으로는 자연스러운 이미지를 얻을 수 없다. 이어 붙여질 이미지의 경계가 서로 많이 다를 수 있기 때문이다.



위와 같은 문제를 해결하고 그림 (c)의 이미지같이 더욱 자연스러운 이미지를 얻기 위하여 아래와 같은 방법(minimum error boundary cut)을 사용한다. 이 문제에서는 편의상 높이가 같은 두 흑백 이미지를 좌우로 합치는 것만 고려한다.

#### <두 이미지를 포개어 자연스러운 경계를 선택하는 과정>

이어 붙일 두 이미지를 B1과 B2라고 하자. (B1은 왼쪽, B2는 오른쪽)

두 이미지의 경계를 조금 포갠다. 포갠 영역의 너비는 최소 2픽셀이다.

포개어진 영역에서 B1과 B2 이미지의 차이가 최소가 되도록 경계선을 결정한다. 그리고 B1 이미지에서 경계선과 그 오른쪽 부분을 B2 이미지를 덮어서 새로운 이미지를 생성한다.

경계선은 포개진 영역의 한 행마다 하나의 픽셀을 선택하여 생성한다.

경계선의 각 행에 선택된 픽셀은 바로 위 혹은 아래 행에서 선택된 픽셀과 두 칸 이상 떨어질 수 없다. 따라서 한 경계선 픽셀은 자신의 바로 아래 혹은 위의 행의 경계선 픽셀과 좌우로 -1, 0, +1 칸 차이가 나는 것만 허용된다.

각 행은 경계선으로 선택된 픽셀을 기준으로 그 왼쪽은 B1 이미지의 픽셀을 그대로 사용하고, 경계선의 픽셀과 그 것의 오른쪽 픽셀들은 B2 이미지의 픽셀로 덮어쓰게 된다.

B1	B1	B1	B1	B1	B1	B2	B2	B2	B2
B1	B1	B1	B1	B1	B1	B1	B2	B2	B2
B1	B1	B1	B1	B1	B1	B2	B2	B2	B2
B1	B1	B1	B1	B1	B2	B2	B2	B2	B2
B1	B1	B1	B1	B1	B2	B2	B2	B2	B2

<포개어진 5x10 영역에서 선택된 경계선 예시>

실제로 포개어진 영역에 존재할 수 있는 경계선은 경우의 수가 많기 때문에 이 중에 가장 자연스럽게 두 이미지를 이어 붙일 수 있는 경계를 선택하여야 한다. 경계선의 부자연스러움 정도는 경계선상의 존재하는 각 픽셀 위치에서의 B1 이미지의 픽셀과 B2 이미지의 픽셀의 밝기 값의 차를 제곱하여 모두 더한 값(sum of squared distance, SSD)으로 정의할 수 있다. 물론 가장 자연스러운 경계선은 SSD 값이 가장 낮은 경계선을 의미한다.

0	79	240	50	62	237	0	62	237
10	110	230	16	58	99	16	58	99
9	130	213	25	120	170	9	120	170
30	70	235	90	120	240	30	70	240

<포개어진 영역의 B1 이미지와 B2 이미지, 그리고 최적의 경계선>

이 문제에서는 흑백 영상만을 다루므로 각 픽셀은 0~255의 256개의 정수 밝기 값으로 표현할 수 있다. 위의 예시에서 두 이미지는 세 번째 그림과 같이 경계선을 선택하면 최적이라고 할 수 있으며 이때의 부자연스러움 정도(SSD)는 아래와 같이 계산할 수 있다.

$$SSD = (79-62)^2 + (10-16)^2 + (130-120)^2 + (235-240)^2 = 450$$

두 이미지의 포개어질 영역의 밝기 값들이 주어질 때, 선택할 수 있는 최적의 경계선이 가지는 최소의 부자연스러움 정도(SSD)를 계산하여 출력하는 프로그램을 작성하시오. (실행시간 0.2초 이하 @i7 3.8GHz CPU, 입력 받는 시간 제외)

#### <입력 조건>

- 첫 번째 줄에는 포개어진 두 이미지의 높이  $H$  ( $1 \leq H \leq 100$ ) 와 포개어진 영역의 너비  $W$  ( $1 \leq W \leq 100$ ) 가 공백으로 구분된 두 정수로 주어진다.  $H$ 는 이미지의 행의 수,  $W$ 는 포개어진 영역의 열의 수를 나타내는 자연수이다. 이 문제는 포개어질 영역의 밝기 값만 주어짐에 유의한다. 입력으로 주어지는 두 이미지는 모두  $H$ 행  $W$ 열이다.
- 그 다음  $H$ 줄에 걸쳐서 B1 이미지의 밝기 값이 주어진다. 각 줄은 영상의 행이며  $W$ 개의 픽셀의 색상 값이 0~255 범위의 정수로 주어진다.
- 그 다음  $H$ 줄에 걸쳐서 B2 이미지의 밝기 값이 주어진다. 입력 형식은 위의 B1 이미지와 같다.

### <출력 조건>

첫 번째 줄에 두 이미지에서 선택할 수 있는 경계선이 가지는 최소의 부자연스러운 정도(SSD)를 한 줄에 출력한다.

두 번째 줄에 입력 받은 이후부터 결과 출력까지 걸린 실행시간을 초 단위로 출력한다.

### <입력/출력 예시>

#### 예제 입력 1 복사

```
4 3
0 79 240
10 110 230
9 130 213
30 70 235
50 62 237
16 58 99
25 120 170
90 120 240
```

#### 예제 출력 1 복사

```
450
실행시간: xxx초
```

## <두 번째 문제 제시> 이미지 풀링 (5점)

전남이는 최근 딥러닝을 공부하던 중, 이미지 처리에 흔히 쓰이는 합성곱 신경망(Convolutional Neural Network, CNN)의 풀링 연산에 영감을 받아 자신만의 풀링을 만들고 이를 222-풀링이라 부르기로 했다.

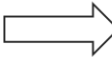
다음은 8×8 행렬이 주어졌다고 가정했을 때 222-풀링을 1회 적용하는 과정을 설명한 것이다

1. 행렬을 2×2 정사각형으로 나눈다.

-1	2	14	7	4	-5	8	9
10	6	23	2	-1	-1	7	11
9	3	5	-2	4	4	6	6
7	15	0	8	21	20	6	6
19	8	12	-8	4	5	2	9
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24

2. 2×2 정사각형에서 2번째로 큰 수만 남긴다.

-1	2	14	7	4	-5	8	9
10	6	23	2	-1	-1	7	11
9	3	5	-2	4	4	6	6
7	15	0	8	21	20	6	6
19	8	12	-8	4	5	2	9
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24



6	14	-1	9
9	5	20	6
8	4	5	8
17	19	21	23

N×N 행렬에 222-풀링을 반복해서 적용하여 크기를 1×1로 만들었을 때 무슨 값이 남아있을지 궁금해 하는 전남이를 위하여 이 문제를 해결해 보자. (실행시간 0.2초 이하 @i7 3.8GHz CPU, 입력 받는 시간 제외)

### <입력 조건>

첫 번째 줄에  $N(2 \leq N \leq 1024)$ 이 주어진다. N은 항상 2의 거듭제곱 꼴이다. ( $N=2^K, 1 \leq K \leq 10$ )

그 다음 N개의 줄에 걸쳐서 각 행의 원소 N개가 차례대로 주어진다. (행렬의 모든 성분은 -10,000 이상 10,000 이하의 정수이다.)

### <출력 조건>

첫 번째 줄에 222-풀링 반복 후 마지막에 남은 숫자를 출력한다.

두 번째 줄에 입력 받은 이후부터 결과 출력까지 걸린 실행시간을 초 단위로 출력한다.

### <입력/출력 예시>

#### 예제 입력 1 복사

```
4
-6 -8 7 -4
-5 -5 14 11
11 11 -1 -1
4 9 -2 -4
```

#### 예제 출력 1 복사

```
11
실행시간: xxx초
```

#### 예제 입력 2 복사

```
8
-1 2 14 7 4 -5 8 9
10 6 23 2 -1 -1 7 11
9 3 5 -2 4 4 6 6
7 15 0 8 21 20 6 6
19 8 12 -8 4 5 2 9
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
```

#### 예제 출력 2 복사

```
17
실행시간: xxx초
```

### <주의사항>

- Cpp 파일 2개(이미지퀵팅.cpp, 이미지풀링.cpp)를 과제 게시판에 업로드할 것 (그 외 파일은 허용 안됨, 압축파일 형태로 제출하지 말 것)
- Cpp 파일의 코드에 주석을 상세히 기입할 것
- 필요한 모든 헤더 파일 및 함수를 cpp 파일에 포함시킬 것
- 띄어쓰기나 줄 바꿈에 주의할 것
- 수강생들간의 Copy 발견 시 모두 0점 처리함
- GNU Compiler Collection (g++ 9.2, clang++ 10.0) 컴파일러에서 에러 없이 실행되어야 함.
- 실행시간 측정 방법, 컴파일러 설치 및 설정 방법은 첫 번째 프로젝트 첨부파일 참조

### <평가기준>

- 이미지퀵팅, 이미지풀링 문제에 대한 배점은 각 5점 만점으로 한다. (총 10점)
- 각 문제에 대한 평가 기준은 다음과 같이 정한다.
- 다양한 입력 테스트케이스에 대해서 프로그램 실행 시 출력 값이 모두 맞고, 실행시간 조건을 만족할 경우 5점 만점 처리함
- 다양한 테스트케이스에 대해서 프로그램 실행 시 출력 값이 모두 맞고, 실행시간 조건을 위반하거나 실행시간 출력이 없는 경우 4점 처리함
- 다양한 테스트케이스에 대해서 프로그램 실행 시 출력 값이 한 번이라도 틀린 경우 3점 처리함
- 컴파일 에러, 런타임 에러 등으로 인해 프로그램 실행이 안 될 경우 2점 처리함
- 주석 설명이 없거나 불충분하면 2점 처리함
- 핵심 구현 내용 부재 시 1점 처리함 (예시: 입력만 받고 처리에 대한 구현이 없는 경우)
- 기한 내 미제출하거나 Copy 발견시 0점 처리함