



Dequeue를 사용해 중첩 부분은 놔두고 바뀌는 부분만 빼고, 넣고 하여 구현한 결과

```
import time
```

```
start = time.time()
```

```
filtered = np.zeros(noisy_face.shape)
```

```
filtered
```

```
windowSize = 3
```

```
gap = windowSize // 2 # 제외되는 픽셀수
```

이동할때 양 끝 각각에
제외되는 픽셀의 수.

```
for i in range(gap, noisy_face.shape[0]-gap):
```

```
# 좌에서 우로 이동하고, 끝이 되면 한칸 내려서 큐를 리셋하고 진행한다.
```

```
dq = Dequeue()
```

```
# 초기 큐를 생성(추가순서는 빼기 쉽게 세로우선으로 추가한다.)
```

```
for c in range(-gap, gap+1):
```

```
    for r in range(-gap, gap+1):
```

```
        dq.insertLast(noisy_face[i+r, 1+c])
```

```
        filtered[i, 1] = np.median(dq, dq)
```

```
# 생성된 큐에서 윈도우 개수 만큼 빼고, 다시 윈도우 개수 만큼 추가한다.
```

```
for j in range(2, noisy_face.shape[0]-1):
```

```
    for k in range(gap, -gap-1, -1):
```

```
        dq.popFirst() # 하나빼고 하나 넣고 ...
```

```
        dq.insertLast(noisy_face[i+k, j+gap])
```

```
        filtered[i, j] = np.median(dq, dq)
```

```
print(time.time()-start) # 계산 시간 프린트
```



16.71644687652588