



# 알고리즘 Week 9

---

20기 정규세션

TOBIG'S 19기 임서영

# Contents

---



15기 정규세션  
TOBIG'S 14기 000

---

Unit 01 | 7주차 리뷰

---

Unit 02 | Greedy 알고리즘

---

Unit 03 | Greedy 알고리즘 문제

---

Unit 04 | 9주차 과제

---



15기 정규세션  
TOBIG'S 14기 000

# Unit 01 | 7주차 리뷰

# Unit 01 | 7주차 리뷰



20기 정규세션  
19기 임서영

## 예제 입력 1 복사

```
5
3 4
1 1
1 -1
2 2
3 3
```

## 예제 출력 1 복사

```
1 -1
1 1
2 2
3 3
3 4
```

## 문제

2차원 평면 위의 점  $N$ 개가 주어진다. 좌표를  $x$ 좌표가 증가하는 순으로,  $x$ 좌표가 같으면  $y$ 좌표가 증가하는 순서로 정렬한 다음 출력하는 프로그램을 작성하시오.

## 입력

첫째 줄에 점의 개수  $N$  ( $1 \leq N \leq 100,000$ )이 주어진다. 둘째 줄부터  $N$ 개의 줄에는  $i$ 번점의 위치  $x_i$ 와  $y_i$ 가 주어진다. ( $-100,000 \leq x_i, y_i \leq 100,000$ ) 좌표는 항상 정수이고, 위치가 같은 두 점은 없다.

## 출력

첫째 줄부터  $N$ 개의 줄에 점을 정렬한 결과를 출력한다.

# Unit 01 | 7주차 리뷰



20기 정규세션  
19기 임서영

- 1) N으로 input을 받기!
- 2) points 라는 리스트를 생성!
- 3) X, y 좌표를 map으로 int 정수로 각각 입력받기!
- 4) 오름차순으로 정리
- 5) Points 리스트 출력!

```
N = int(input())

# 리스트 생성
points = []

# x, y를 tuple로 저장함
for _ in range(N):
    x, y = map(int, input().split())
    points.append((x, y))

# 정리하기 (x increasing, y increasing)
points.sort()

# Output the sorted points
for x, y in points:
    print(x, y)
```

# Unit 01 | 7주차 리뷰



20기 정규세션  
19기 임시영

아래 <그림 1>과 같이 여러개의 정사각형칸들로 이루어진 정사각형 모양의 종이가 주어져 있고, 각 정사각형들은 하얀색으로 칠해져 있거나 파란색으로 칠해져 있다. 주어진 종이를 일정한 규칙에 따라 잘라서 다양한 크기를 가진 정사각형 모양의 하얀색 또는 파란색 색종이를 만들려고 한다.

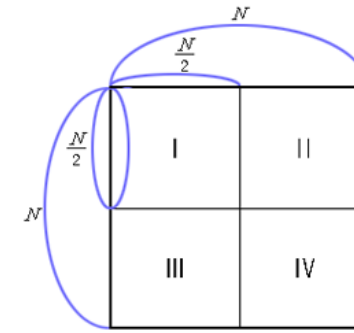
1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 1> 8×8 종이

전체 종이의 크기가  $N \times N$  ( $N=2^k$ ,  $k$ 는 1 이상 7 이하의 자연수) 이라면 종이를 자르는 규칙은 다음과 같다.

전체 종이가 모두 같은 색으로 칠해져 있지 않으면 가로와 세로로 중간 부분을 잘라서 <그림 2>의 I, II, III, IV와 같이 똑같은 크기의 네 개의  $N/2 \times N/2$  색종이로 나눈다. 나누어진 종이 I, II, III, IV 각각에 대해서도 앞에서와 마찬가지로 모두 같은 색으로 칠해져 있지 않으면 같은 방법으로 똑같은 크기의 네 개의 색종이로 나눈다. 이와 같은 과정을 잘라진 종이가 모두 하얀색 또는 모두 파란색으로 칠해져 있거나, 하나의 정사각형 칸이 되어 더 이상 자를 수 없을 때까지 반복한다.

위와 같은 규칙에 따라 잘랐을 때 <그림 3>은 <그림 1>의 종이를 처음 나눈 후의 상태를, <그림 4>는 두 번째 나눈 후의 상태를, <그림 5>는 최종적으로 만들어진 다양한 크기의 9장의 하얀색 색종이와 7장의 파란색 색종이를 보여주고 있다.



<그림 2>

1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 3> 처음 나눈 후의 상태

1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 4> 두 번째 나눈 후의 상태

1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 5> 최종적으로 나누어진 색종이들

입력으로 주어진 종이의 한 번의 길이  $N$ 과 각 정사각형칸의 색(하얀색 또는 파란색)이 주어질 때 잘라진 하얀색 색종이와 파란색 색종이의 개수를 구하는 프로그램을 작성하시오.

# Unit 01 | 7주차 리뷰



20기 정규세션  
19기 임서영

White : 0 , Blue 1 (변수를 잡아두기)

- 1) 색상 체크가 포인트 - paper에서 잘라진 모든 영역들의 색상이 동일한지 확인할것
- 2) Check 영역에서- 그 영역에서 같은 색이면 카운터 증가시킬것 !

```
def cut_paper(x, y, size):
    global white, blue
    check = paper[x][y]

    for i in range(x, x + size):
        for j in range(y, y + size):
            if paper[i][j] != check:
                cut_paper(x, y, size // 2) # Top-left
                cut_paper(x, y + size // 2, size // 2) # Top-right
                cut_paper(x + size // 2, y, size // 2) # Bottom-left
                cut_paper(x + size // 2, y + size // 2, size // 2) # Bottom-right
            return

    if check == 0:
        white += 1
    else:
        blue += 1

N = int(input())
paper = [list(map(int, input().split())) for _ in range(N)]
white, blue = 0, 0

cut_paper(0, 0, N)

print(white)
print(blue)
```

# Unit 01 | 7주차 리뷰



20기 정규세션  
19기 임서영

White : 0 , Blue 1 (변수를 잡아두기)

1) 색상 체크가 포인트 – paper에서 잘라진  
모든 영역들의 색상이 동일한지 확인할것

2) Check 영역에서- 그 영역에서 같은 색이면  
카운터 증가시킬것 !

1	1	0	0
1	1	0	0
0	0	1	1
0	0	1	1

예시 :  $N = 4$

- 좌상단: 모든 셀이 1로 동일하므로 파란색 카운터를 1 증가.
- 우상단: 모든 셀이 0로 동일하므로 흰색 카운터를 1 증가.
- 좌하단: 모든 셀이 0로 동일하므로 흰색 카운터를 1 증가.
- 우하단: 모든 셀이 1로 동일하므로 파란색 카운터를 1 증가.



# Unit 01 | 7주차 리뷰

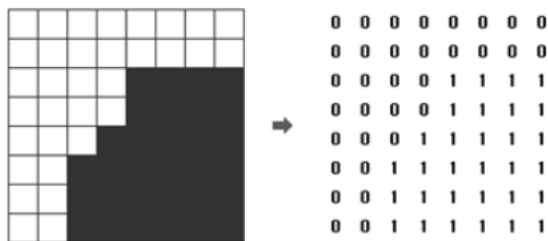


20기 정규세션  
19기 임시영

## 문제

흑백 영상을 압축하여 표현하는 데이터 구조로 쿼드 트리(Quad Tree)라는 방법이 있다. 흰 점을 나타내는 0과 검은 점을 나타내는 1로만 이루어진 영상(2차원 배열)에서 같은 숫자의 점들이 한 곳에 많이 몰려있으면, 쿼드 트리에서는 이를 압축하여 간단히 표현할 수 있다.

주어진 영상이 모두 0으로만 되어 있으면 압축 결과는 "0"이 되고, 모두 1로만 되어 있으면 압축 결과는 "1"이 된다. 만약 0과 1이 섞여 있으면 전체를 한 번에 나타내지를 못하고, 왼쪽 위, 오른쪽 위, 왼쪽 아래, 오른쪽 아래, 이렇게 4개의 영상으로 나누어 압축하게 되며, 이 4개의 영역을 압축한 결과를 차례대로 괄호 안에 묶어서 표현한다



위 그림에서 왼쪽의 영상은 오른쪽의 배열과 같이 숫자로 주어지며, 이 영상을 쿼드 트리 구조를 이용하여 압축하면 "(0(0011)(0(0111)01)1)"로 표현된다.  $N \times N$  크기의 영상이 주어질 때, 이 영상을 압축한 결과를 출력하는 프로그램을 작성하시오.

## 입력

첫째 줄에는 영상의 크기를 나타내는 숫자  $N$  이 주어진다.  $N$  은 언제나 2의 제곱수로 주어지며,  $1 \leq N \leq 64$ 의 범위를 가진다. 두 번째 줄부터는 길이  $N$ 의 문자열이  $N$ 개 들어온다. 각 문자열은 0 또는 1의 숫자로 이루어져 있으며, 영상의 각 점들을 나타낸다.

## 출력

영상을 압축한 결과를 출력한다.

### 예제 입력 1 복사

```
8
11110000
11110000
00011100
00011100
11110000
11110000
11110011
11110011
```

### 예제 출력 1 복사

```
((110(0101))(0010)1(0001))
```

# Unit 01 | 7주차 리뷰



20기 정규세션  
19기 임서영

```
def compress_quad_tree(image):  
    N = len(image)
```

```
    if N == 1:  
        return image[0][0]
```

1) 이미지 크기가 1인 경우 ,  
픽셀값 0 or 1출력할것

```
    # 0, 1만 있는지 확인
```

```
    is_uniform = True
```

```
    for i in range(N):
```

2) 동일한 픽셀인지 확인

```
        for j in range(N):  
            if image[i][j] != image[0][0]:  
                is_uniform = False  
                break
```

```
    if is_uniform:  
        return str(image[0][0])  
    else:
```

3) 모든 픽셀이 동일하면 그 값 출력

```
    # 4개로 나눔!
```

```
    half = N // 2
```

```
    upper_left = [row[:half] for row in image[:half]]
```

```
    upper_right = [row[half:] for row in image[:half]]
```

```
    lower_left = [row[:half] for row in image[half:]]
```

```
    lower_right = [row[half:] for row in image[half:]]
```

4) 영역 나누기

```
    # compress 시키기 , 각각 boundary 별로 !
```

```
    compressed_upper_left = compress_quad_tree(upper_left)
```

```
    compressed_upper_right = compress_quad_tree(upper_right)
```

```
    compressed_lower_left = compress_quad_tree(lower_left)
```

```
    compressed_lower_right = compress_quad_tree(lower_right)
```

```
    # 나누것을 합치기
```

```
    result = "(" + compressed_upper_left + compressed_upper_right + compressed_lower_left + compressed_lower_right +
```

```
    return result
```

# Unit 01 | 7주차 리뷰



20기 정규세션  
19기 임서영

```
# 나눈것을 합치기
result = "(" + compressed_upper_left + compressed_upper_right + compressed_lower_left + compressed_lower_right

return result

# Read input
N = int(input())
image = [list(input()) for _ in range(N)]

# quad tree 만들기
compressed_image = compress_quad_tree(image)

# Print the compressed image
print(compressed_image)
```

```
0010
0010
1100
1100
```

예시

- 좌상단: 00 / 00
- 우상단: 10 / 10
- 좌하단: 11 / 11
- 우하단: 00 / 00

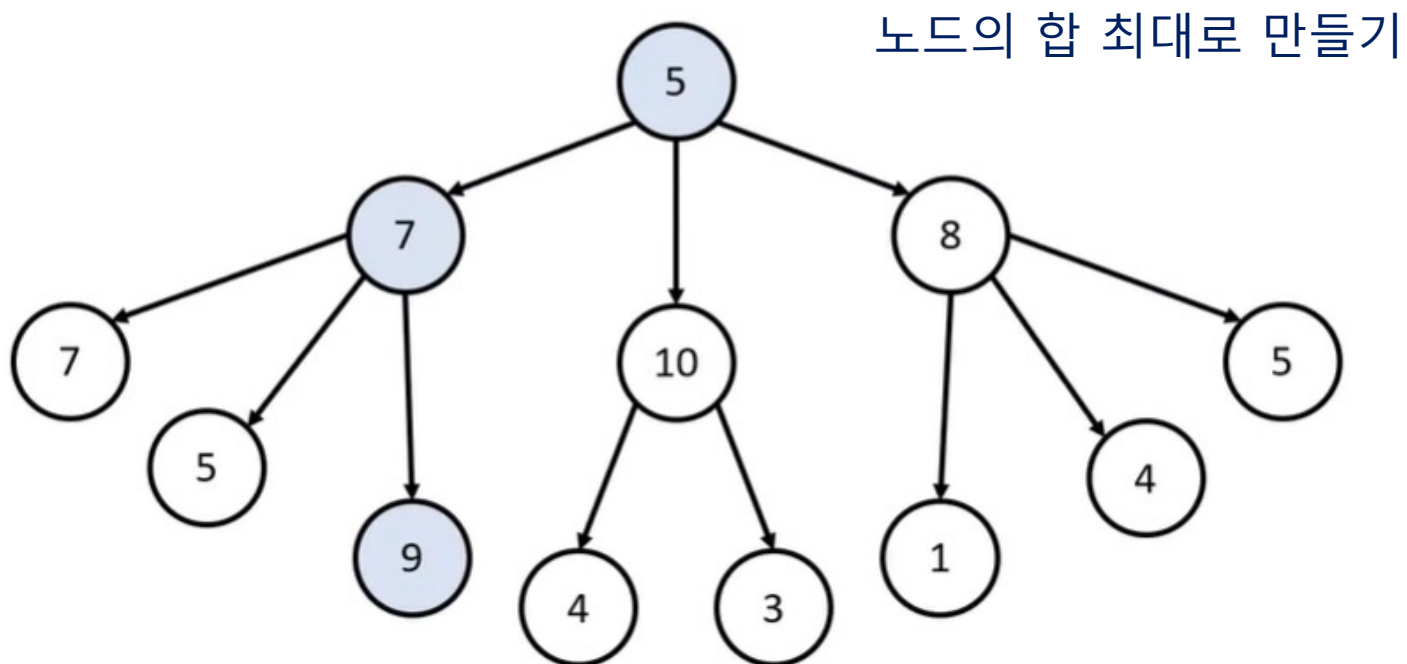
## 압축결과

- 좌상단: 0
- 우상단: (0)(1)(0)(1) => (0101)
- 좌하단: 1
- 우하단: 0

정답!

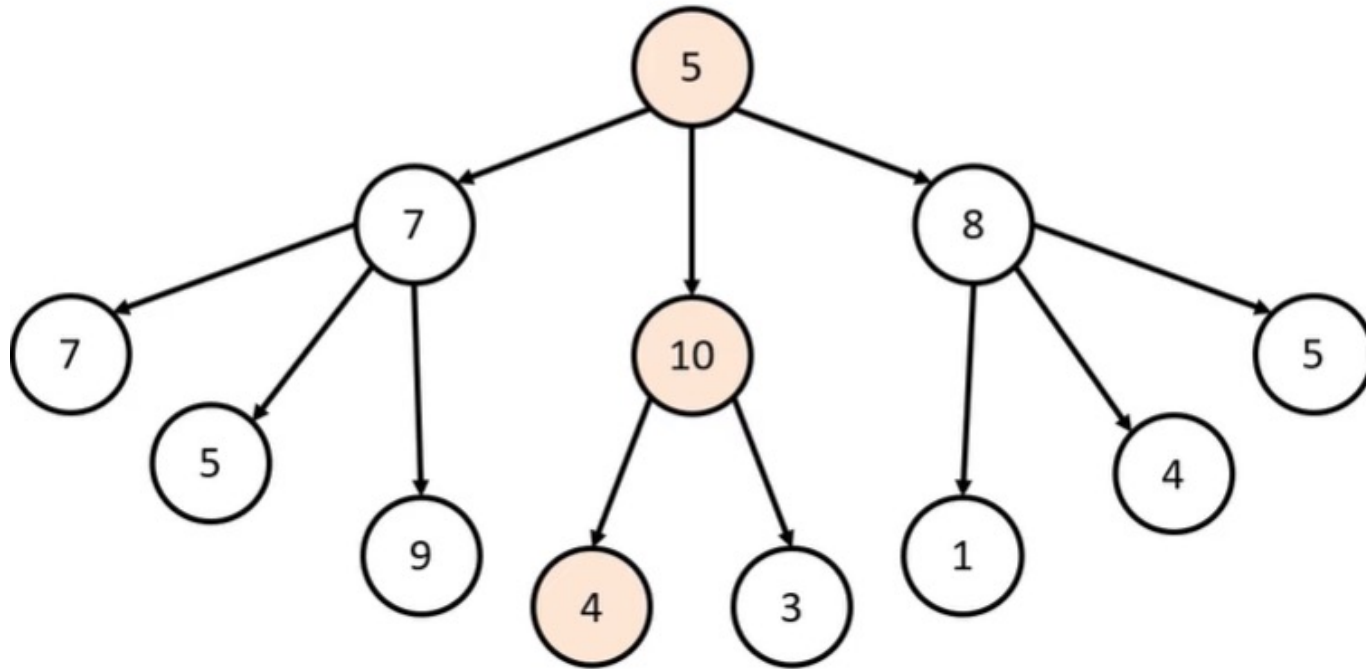
**(0)(0101)(1)(0) => (0(0101)10)**

## 그리디 알고리즘 (GREEDY ALGORITHM)



## 그리디 알고리즘 (GREEDY ALGORITHM)

매 상황에서 가장 큰 값만 고르는것 !



## 그리디 알고리즘 (GREEDY ALGORITHM)

### 1) 탐욕 알고리즘을 해결하는 방법

- 선택 절차 (Selection Procedure) : **현재 상황에서** 최적의 해답을 구할것
- 적절성 검사 : 선택된 해가 문제의 조건을 만족하는 검사
- 해답 검사: 원래의 문제가 해결되었는지 검사하고 , 아니라면 **반복할것**

### 2) 단, 문제가 2가지 조건을 성립해야함

- 앞의 선택이 **이후의 선택에** 영향을 주지 않아야함
- 문제에 대한 최종 해결 방법은 부분 문제에 대한 최적의 해결 방법이어야 함

# Unit 04 | 9주차 과제



20기 정규세션  
19기 임서영

동전 0 - <https://www.acmicpc.net/problem/11047>

## 문제

준규가 가지고 있는 동전은 총  $N$ 종류이고, 각각의 동전을 매우 많이 가지고 있다.

동전을 적절히 사용해서 그 가치의 합을  $K$ 로 만들려고 한다. 이때 필요한 동전 개수의 최솟값을 구하는 프로그램을 작성하시오.

## 입력

첫째 줄에  $N$ 과  $K$ 가 주어진다. ( $1 \leq N \leq 10$ ,  $1 \leq K \leq 100,000,000$ )

둘째 줄부터  $N$ 개의 줄에 동전의 가치  $A_i$ 가 오름차순으로 주어진다. ( $1 \leq A_i \leq 1,000,000$ ,  $A_1 = 1$ ,  $i \geq 2$ 인 경우에  $A_i$ 는  $A_{i-1}$ 의 배수)

## 출력

첫째 줄에  $K$ 원을 만드는데 필요한 동전 개수의 최솟값을 출력한다.

## 예제 입력 1 복사

```
10 4200
1
5
10
50
100
500
1000
5000
10000
50000
```

# Unit 04 | 9주차 과제



20기 정규세션  
19기 임서영

강의실 배정 - <https://www.acmicpc.net/problem/11000>

## 강의실 배정



시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	38451	11290	8331	28.925%

## 문제

수강신청의 마스터 김종혜 선생님에게 새로운 과제가 주어졌다.

김종혜 선생님한테는  $S_i$ 에 시작해서  $T_i$ 에 끝나는  $N$ 개의 수업이 주어지는데, 최소의 강의실을 사용해서 모든 수업을 가능하게 해야 한다.

참고로, 수업이 끝난 직후에 다음 수업을 시작할 수 있다. (즉,  $T_i \leq S_j$  일 경우  $i$  수업과  $j$  수업은 같이 들을 수 있다.)

수강신청 대충한 게 찢리면, 선생님을 도와드리자!

## 입력

첫 번째 줄에  $N$ 이 주어진다. ( $1 \leq N \leq 200,000$ )

이후  $N$ 개의 줄에  $S_i, T_i$ 가 주어진다. ( $0 \leq S_i < T_i \leq 10^9$ )

## 예제 입력 1 복사

```
3
1 3
2 4
3 5
```



# Unit 04 | 9주차 과제



20기 정규세션  
19기 임서영

신입 사원 - <https://www.acmicpc.net/problem/1946>

## 문제

언제나 최고만을 지향하는 굴지의 대기업 진영 주식회사가 신규 사원 채용을 실시한다. 인재 선발 시험은 1차 서류심사와 2차 면접시험으로 이루어진다. 최고만을 지향한다는 기업의 이념에 따라 그들은 최고의 인재들만을 사원으로 선발하고 싶어 한다.

그래서 진영 주식회사는, 다른 모든 지원자와 비교했을 때 서류심사 성적과 면접시험 성적 중 적어도 하나가 다른 지원자보다 떨어지지 않는 자만 선발한다는 원칙을 세웠다. 즉, 어떤 지원자 A의 성적이 다른 어떤 지원자 B의 성적에 비해 서류 심사 결과와 면접 성적이 모두 떨어진다면 A는 결코 선발되지 않는다.

이러한 조건을 만족시키면서, 진영 주식회사가 이번 신규 사원 채용에서 선발할 수 있는 신입사원의 최대 인원수를 구하는 프로그램을 작성하시오.

## 입력

첫째 줄에는 테스트 케이스의 개수  $T(1 \leq T \leq 20)$ 가 주어진다. 각 테스트 케이스의 첫째 줄에 지원자의 숫자  $N(1 \leq N \leq 100,000)$ 이 주어진다. 둘째 줄부터  $N$ 개 줄에는 각각의 지원자의 서류심사 성적, 면접 성적의 순위가 공백을 사이에 두고 한 줄에 주어진다. 두 성적 순위는 모두 1위부터  $N$ 위까지 동석차 없이 결정된다고 가정한다.

## 출력

각 테스트 케이스에 대해서 진영 주식회사가 선발할 수 있는 신입사원의 최대 인원수를 한 줄에 하나씩 출력한다.

## 예제 입력 1 복사

```
2
5
3 2
1 4
4 1
2 3
5 5
7
3 6
7 3
4 2
1 4
5 7
2 5
6 1
```

# Unit 04 | 9주차 과제



20기 정규세션  
19기 임시영

## 멀티탭 스케줄링 - <https://www.acmicpc.net/problem/1700>

### 문제

기숙사에서 살고 있는 준규는 한 개의 멀티탭을 이용하고 있다. 준규는 키보드, 헤어드라이기, 핸드폰 충전기, 디지털 카메라 충전기 등 여러 개의 전기용품을 사용하면서 어쩔 수 없이 각종 전기용품의 플러그를 뺐다 꽂았다 하는 불편함을 겪고 있다. 그래서 준규는 자신의 생활 패턴을 분석하여, 자기가 사용하고 있는 전기용품의 사용순서를 알아내었고, 이를 기반으로 플러그를 빼는 횟수를 최소화하는 방법을 고안하여 보다 쾌적한 생활환경을 만들려고 한다.

예를 들어 3 구(구멍이 세 개 달린) 멀티탭을 쓸 때, 전기용품의 사용 순서가 아래와 같이 주어진다면,

1. 키보드
2. 헤어드라이기
3. 핸드폰 충전기
4. 디지털 카메라 충전기
5. 키보드
6. 헤어드라이기

키보드, 헤어드라이기, 핸드폰 충전기의 플러그를 순서대로 멀티탭에 꽂은 다음 디지털 카메라 충전기 플러그를 꽂기 전에 핸드폰충전기 플러그를 빼는 것이 최적일 것이므로 플러그는 한 번만 빼면 된다.

### 입력

첫 줄에는 멀티탭 구멍의 개수  $N$  ( $1 \leq N \leq 100$ )과 전기 용품의 총 사용횟수  $K$  ( $1 \leq K \leq 100$ )가 정수로 주어진다. 두 번째 줄에는 전기용품의 이름이  $K$  이하의 자연수로 사용 순서대로 주어진다. 각 줄의 모든 정수 사이는 공백문자로 구분되어 있다.

### 출력

하나씩 플러그를 빼는 최소의 횟수를 출력하시오.

### 예제 입력 1 복사

```
2 7
2 3 2 3 1 2 7
```



20기 정규세션  
19기 임서영

들어주셔서 감사합니다.

TOBIG'S