

# Chapter 01.

## 기본 자료구조

### 핵심 유형 문제풀이

핵심 유형 문제풀이 | 다양한 문제를 접하며 코딩 테스트에 익숙해지기

강사 나동빈

# Chapter 01. 기본 자료구조

핵심 유형 문제풀이

## Ch1. 기본 자료구조 혼자 힘으로 풀어보기

핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

문제 제목: 풍선 터뜨리기

문제 난이도: 하(Easy)

문제 유형: 자료구조, 덱

추천 풀이 시간: 30분

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 양방향 순환 큐의 기능은 덱(deque)을 이용해 대체할 수 있습니다.
  1. 회전 연산 (왼쪽으로 돌리기): *popleft()* - *append()*
  2. 회전 연산 (오른쪽으로 돌리기): *pop()* - *appendleft()*
  3. 추출 연산 (앞에서 꺼내기): *popleft()*

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

### [알고리즘 요약]

- 풍선에 적힌 원소를 모두 덱(deque)에 삽입한다.
- 가장 앞에 있는 풍선(원소)을 꺼내어 거기에 적힌 수를 확인한다.
- 큐가 빌 때까지 다음을 반복 수행한다.
  - 만약 양수라면  $X - 1$ 만큼 왼쪽으로 돌린다.
  - 만약 음수라면  $|X|$ 만큼 오른쪽으로 돌린다.
  - 이후에 원소를 꺼낸다.
- 단, 최종적으로 풍선의 번호를 출력해야 하므로, (값, 번호) 형태로 원소를 덱(deque)에 삽입해야 한다.

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 아이디어: **덱(deque)**을 이용해 원소의 회전 기능을 구현할 수 있습니다.
- 풍선 목록: [3, 2, 1, -3, -1] / 덱에 원소를 삽입할 때 (값, 번호) 형태로 삽입합니다.

① 꺼낸 원소: []

덱의 상태: [3, 2, 1, -3, -1]

② 꺼낸 원소: [3]

덱의 상태: [-3, -1, 2, 1]

③ 꺼낸 원소: [3, -3]

덱의 상태: [-1, 2, 1]

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 아이디어: **덱(deque)**을 이용해 원소의 회전 기능을 구현할 수 있습니다.
- 풍선 목록: [3, 2, 1, -3, -1] / 덱에 원소를 삽입할 때 (값, 번호) 형태로 삽입합니다.

④ 꺼낸 원소: [3, -3, -1]

덱의 상태: [1, 2]

⑤ 꺼낸 원소: [3, -3, -1, 1]

덱의 상태: [2]

⑥ 꺼낸 원소: [3, -3, -1, 1, 2]

덱의 상태: []

실제로 결과를 출력할 때는 각 **풍선의 번호**에 해당하는 [1, 4, 5, 3, 2]를 출력합니다.

## Ch1. 기본 자료구조 소스 코드

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

```
import sys
# 빠른 입력 함수 사용
input = sys.stdin.readline
from collections import deque

n = int(input()) # 원소의 개수 N
# 전체 원소 리스트
arr = list(map(int, input().split()))
d = deque() # 덱(deque) 초기화
for i in range(n):
    # (수, 번호) 형태로 원소를 삽입
    d.append((arr[i], i + 1))
result = [] # 결과 배열
```

```
current, index = d.popleft() # 원소 추출
result.append(index)
for i in range(n - 1): # 원소를 모두 꺼내기
    if current > 0: # 양수라면
        # current - 1번 "왼쪽으로 돌리기" 수행
        for j in range(current - 1):
            x = d.popleft()
            d.append(x)
    else: # 음수라면 (0은 없음)
        # |current|번 "오른쪽으로 돌리기" 수행
        for j in range(-current):
            x = d.pop()
            d.appendleft(x)
    # 원소 추출
    current, index = d.popleft()
    result.append(index)

for x in result: # 결과 출력
    print(x, end=' ')
```



## Ch1. 기본 자료구조    혼자 힘으로 풀어보기

핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

문제 제목: 트럭

문제 난이도: 중(Medium)

문제 유형: 자료구조, 큐

추천 풀이 시간: 40분

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 단순히 큐를 이용하여 전체 과정을 시뮬레이션하여 문제를 해결할 수 있다.
  1. 큐(다리)가 가득 찼으면 큐에서 원소 하나를 추출한다.
  - 2-1) 트럭 배열에서 하나씩 트럭을 꺼내서 큐에 삽입한다.
  - 2-2) 이때 만약에 트럭을 큐에 넣을 수 없다면(하중이 초과된다면) **0**을 삽입한다.

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 아이디어: 큐(queue)를 이용해 전체 과정을 시뮬레이션하여 문제를 해결합니다.
- 다리의 길이  $W=2$ , 최대 하중  $L=10$ , 트럭의 무게=[7, 4, 5, 6]

① 초기에는 큐가 비어 있다.

트럭 배열: [7, 4, 5, 6] / 큐(다리): [] (총 무게: 0)

② 7을 큐에 삽입한다.

트럭 배열: [4, 5, 6] / 큐(다리): [7] (총 무게: 7)

③ 다음 트럭을 넣으면 하중이 초과되므로, 0을 큐에 삽입한다.

트럭 배열: [4, 5, 6] / 큐(다리): [7, 0] (총 무게: 7)

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 아이디어: 큐(queue)를 이용해 전체 과정을 시뮬레이션하여 문제를 해결합니다.

- 다리의 길이  $W=2$ , 최대 하중  $L=10$ , 트럭의 무게=[7, 4, 5, 6]

④ 큐가 가득 찼으므로 7을 꺼내고, 4를 삽입한다.

트럭 배열: [5, 6] / 큐(다리): [0, 4] (총 무게: 4)

⑤ 큐가 가득 찼으므로 0을 꺼내고, 5를 삽입한다.

트럭 배열: [6] / 큐(다리): [4, 5] (총 무게: 9)

⑥ 큐가 가득 찼으므로 4를 꺼내고, 6을 넣지 못하므로 0을 삽입한다.

트럭 배열: [6] / 큐(다리): [5, 0] (총 무게: 5)

## Ch1. 기본 자료구조 문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 아이디어: 큐(queue)를 이용해 전체 과정을 시뮬레이션하여 문제를 해결합니다.
- 다리의 길이  $W=2$ , 최대 하중  $L=10$ , 트럭의 무게=[7, 4, 5, 6]

⑦ 큐가 가득 찼으므로 5를 꺼내고, 6을 삽입한다.  
트럭 배열: [] / 큐(다리): [0, 6] (총 무게: 6)

트럭 배열이 비었고, 총 무게가 0이므로  
종료합니다. (정답: 8초)

⑧ 큐가 가득 찼으므로 0을 꺼내고, 0을 삽입한다.  
트럭 배열: [] / 큐(다리): [6, 0] (총 무게: 6)

⑨ 큐가 가득 찼으므로 0을 꺼내고, 0을 삽입한다.  
트럭 배열: [] / 큐(다리): [0, 0] (총 무게: 0)

## Ch1. 기본 자료구조    소스 코드

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

```
import sys
# 빠른 입력 함수 사용
input = sys.stdin.readline
from collections import deque

# 트럭의 수, 다리의 길이, 최대 하중
n, w, l = map(int, input().split())
trucks = list(map(int, input().split()))
trucks.reverse() # 트럭 배열
# 덱(deque)을 큐 목적으로 사용
q = deque()
total = 0 # 총 무게
t = 0 # 총 소요 시간
```

```
while True:
    # 모든 트럭을 처리한 경우 종료
    if len(trucks) == 0 and total == 0:
        break
    if len(q) == w: # 큐가 가득 찬 경우 꺼내기
        x = q.popleft()
        total -= x
    # 트럭이 들어갈 수 있는 경우
    if len(trucks) > 0 and total + trucks[-1] <= l:
        q.append(trucks[-1]) # 큐(다리)에 추가
        total += trucks[-1] # 무게 반영
        trucks.pop()
    else: # 트럭이 들어가지 못 한다면 0을 삽입
        q.append(0)
    t += 1

print(t) # 총 소요 시간 출력
```

## Ch1. 기본 자료구조    혼자 힘으로 풀어보기

핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

문제 제목: 나는 요리사다

문제 난이도: 하(Easy)

문제 유형: 문법, 배열, 이차원 배열

추천 풀이 시간: 10분

## Ch1. 기본 자료구조    문제 풀이 핵심 아이디어

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

- 2차원 배열을 다룰 수 있다면 해결할 수 있는 문제입니다.
- 문제 해결을 위해 단순히 2차원 배열을 이용하여 각 행(row)마다 합계를 계산합니다.



## Ch1. 기본 자료구조    소스 코드

### 핵심 유형 문제풀이

## Ch1.

핵심 유형 문제풀이

```
result = 0 # 최종 우승자
max_value = 0 # 평가 점수의 최댓값

# 요리사는 총 5명
for i in range(5):
    # 각 요리사마다 하나의 행(row)을 차지
    row = list(map(int, input().split()))
    summary = sum(row) # 현재 요리사의 합
    # 현재까지의 최대 합보다 현재 행(row)의 합이 더 큰 경우
    if max_value < summary:
        max_value = summary
        result = i + 1

print(result, max_value)
```