취업 준비 스터디 4주차 멘토링

2022.10.19

발표자 : 박동학

목차

1. 피보나치 수열

다음 소스는 N번째 피보나치 수를 구하는 C++ 함수이다.

```
int fibonacci(int n) {
    if (n == 0) {
        printf("0");
        return 0;
    } else if (n == 1) {
        printf("1");
        return 1;
    } else {
        return fibonacci(n-1) + fibonacci(n-2);
    }
}
```

`fibonacci(3)` 을 호출하면 다음과 같은 일이 일어난다.

- `fibonacci(3)` 은 `fibonacci(2)` 와 `fibonacci(1)` (첫 번째 호출)을 호출한다.
- `fibonacci(2)` 는 `fibonacci(1)` (두 번째 호출)과 `fibonacci(0)` 을 호출한다.
- 두 번째 호출한 `fibonacci(1)` 은 1을 출력하고 1을 리턴한다.
- `fibonacci(0)` 은 0을 출력하고, 0을 리턴한다.
- `fibonacci(2)` 는 `fibonacci(1)` 과 `fibonacci(0)` 의 결과를 얻고, 1을 리턴한다.
- 첫 번째 호출한 `fibonacci(1)` 은 1을 출력하고, 1을 리턴한다.
- `fibonacci(3)` 은 `fibonacci(2)` 와 `fibonacci(1)` 의 결과를 얻고, 2를 리턴한다.

1은 2번 출력되고, 0은 1번 출력된다. N이 주어졌을 때, `fibonacci(N)` 을 호출했을 때, 0과 1이 각각 몇 번 출력되는지 구하는 프로그램을 작성하시오.

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다.

각 테스트 케이스는 한 줄로 이루어져 있고, N이 주어진다. N은 40보다 작거나 같은 자연수 또는 0이다.

출력

각 테스트 케이스마다 0이 출력되는 횟수와 1이 출력되는 횟수를 공백으로 구분해서 출력한다.

1. 피보나치 수열

```python
def cal(num):
    length = len(zero)
    if length <= num:
        for i in range(length, num + 1):
            zero.append(zero[i - 1] + zero[i - 2])
            one.append(one[i - 1] + one[i - 2])
    print("{} {}".format(zero[num], one[num]))

TestCase = int(input())

zero = [1, 0, 1]
one = [0, 1, 1]

for T in range(TestCase):
    N = int(input())
    cal(N)
```

2. 유기농 배추

문제

차세대 영농인 한나는 강원도 고랭지에서 유기농 배추를 재배하기로 하였다. 농약을 쓰지 않고 배추를 재배하려면 배추를 해충으로부터 보호하는 것이 중요하기 때문에, 한나는 해충 방지에 효과적인 배추흰지렁이를 구입하기로 결심한다. 이 지렁이는 배추근처에 서식하며 해충을 잡아 먹음으로써 배추를 보호한다. 특히, 어떤 배추에 배추흰지렁이가 한 마리라도 살고 있으면 이 지렁이는 인접한 다른 배추로 이동할 수 있어, 그 배추들 역시 해충으로부터 보호받을 수 있다. 한 배추의 상하좌우 네 방향에 다른 배추가 위치한 경우에 서로 인접해있는 것이다.

한나가 배추를 재배하는 땅은 고르지 못해서 배추를 군데군데 심어 놓았다. 배추들이 모여있는 곳에는 배추흰지렁이가 한 마리만 있으면 되므로 서로 인접해있는 배추들이 몇 군데에 퍼져있는지 조사하면 총 몇 마리의 지렁이가 필요한지 알 수 있다. 예를 들어 배추밭이 아래와 같이 구성되어 있으면 최소 5마리의 배추흰지렁이가 필요하다. 0은 배추가 심어져 있지 않은 땅이고, 1은 배추가 심어져 있는 땅을 나타낸다.

1	**1**	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0	0
0	0	**1**	**1**	0	0	0	**1**	**1**	**1**
0	0	0	0	**1**	0	0	**1**	**1**	**1**

입력

입력의 첫 줄에는 테스트 케이스의 개수 T가 주어진다. 그 다음 줄부터 각각의 테스트 케이스에 대해 첫째 줄에는 배추를 심은 배추밭의 가로길이 M($1 \le M \le 50$)과 세로길이 N($1 \le N \le 50$), 그리고 배추가 심어져 있는 위치의 개수 K($1 \le K \le 2500$)이 주어진다. 그 다음 K줄에는 배추의 위치 X($0 \le X \le M-1$), Y($0 \le Y \le N-1$)가 주어진다. 두 배추의 위치가 같은 경우는 없다.

출력

각 테스트 케이스에 대해 필요한 최소의 배추흰지렁이 마리 수를 출력한다.

2. 유기농 배추

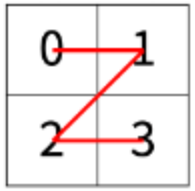
```python
T = int(input())
dx = [1,-1,0,0]
dy = [0,0,1,-1]

Arr = []
visited = []
answer = 0

def bfs(x,y):
    Q = []
    Q.append([x,y])

    while Q:
        x1, y1 = Q.pop(0)
        visited[y1][x1] = True
        for i in range(4):
            nx = x1 + dx[i]
            ny = y1 + dy[i]
            if nx < M and nx >= 0 and ny < N and ny >=0 and visited[ny][nx] is False:
                if [nx,ny] in Arr:
                    visited[ny][nx] = True
                    Q.append([nx,ny])
    answer += 1
    return True
```

2. 유기농 배추

```
27  for tc in range(T):
28      M, N, K = map(int, input().split())
29      for i in range(K):
30          Arr.append(list(map(int, input().split())))
31
32      visited = [ [False for _ in range(M)] for _ in range(N) ]
33
34      for a in range(M):
35          for b in range(N):
36              if [a,b] in Arr:
37                  if visited[b][a] is not True:
38                      bfs(a,b)
39      print(answer)
```

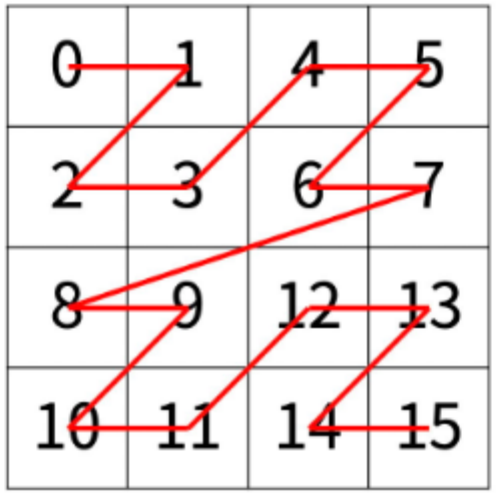
3. Z

한수는 크기가 $2^N \times 2^N$인 2차원 배열을 Z모양으로 탐색하려고 한다. 예를 들어, 2×2배열을 왼쪽 위칸, 오른쪽 위칸, 왼쪽 아래칸, 오른쪽 아래칸 순서대로 방문하면 Z모양이다.



N > 1인 경우, 배열을 크기가 $2^{N-1} \times 2^{N-1}$로 4등분 한 후에 재귀적으로 순서대로 방문한다.

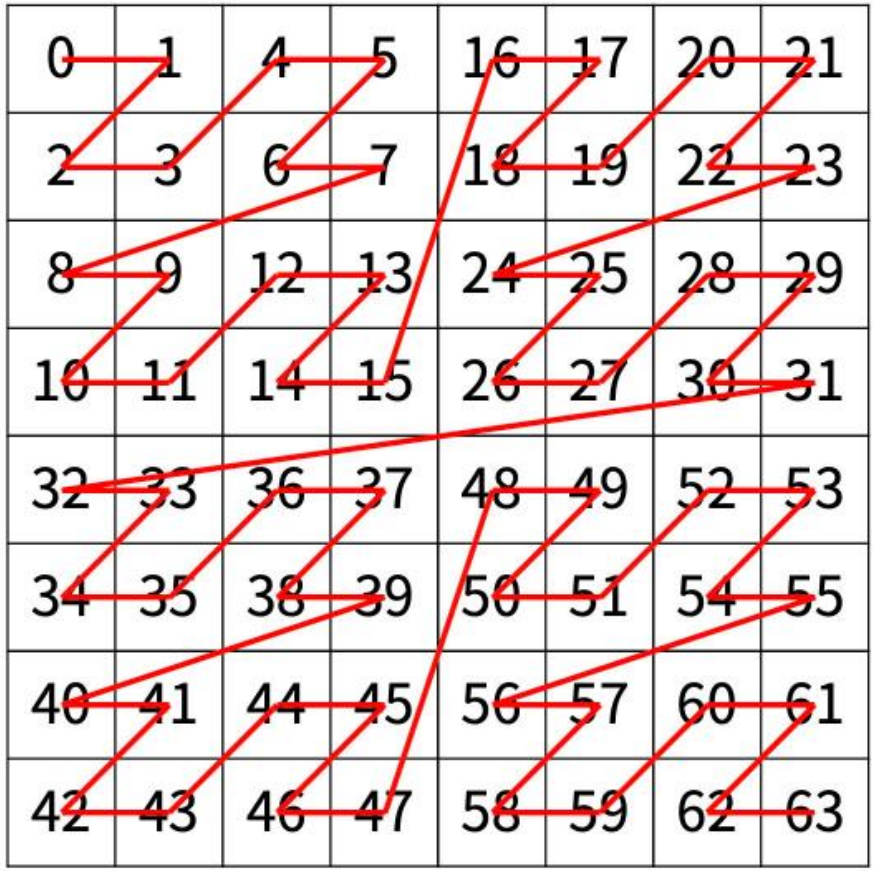
다음 예는 $2^2 \times 2^2$ 크기의 배열을 방문한 순서이다.



N이 주어졌을 때, r행 c열을 몇 번째로 방문하는지 출력하는 프로그램을 작성하시오.

다음은 N=3일 때의 예이다.

3. Z



입력

첫째 줄에 정수 N, r, c가 주어진다.

출력

r행 c열을 몇 번째로 방문했는지 출력한다.

4. 리모컨

리모컨 성공

☆

골드 V

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	77334	18529	12857	22.559%

문제

수빈이는 TV를 보고 있다. 수빈이는 채널을 돌리려고 했지만, 버튼을 너무 세게 누르는 바람에, 일부 숫자 버튼이 고장났다.

리모컨에는 버튼이 0부터 9까지 숫자, +와 -가 있다. +를 누르면 현재 보고있는 채널에서 +1된 채널로 이동하고, -를 누르면 -1된 채널로 이동한다. 채널 0에서 -를 누른 경우에는 채널이 변하지 않고, 채널은 무한대 만큼 있다.

수빈이가 지금 이동하려고 하는 채널은 N이다. 어떤 버튼이 고장났는지 주어졌을 때, 채널 N으로 이동하기 위해서 버튼을 최소 몇 번 눌러야하는지 구하는 프로그램을 작성하시오.

수빈이가 지금 보고 있는 채널은 100번이다.

입력

첫째 줄에 수빈이가 이동하려고 하는 채널 N (0 ≤ N ≤ 500,000)이 주어진다. 둘째 줄에는 고장난 버튼의 개수 M (0 ≤ M ≤ 10)이 주어진다. 고장난 버튼이 있는 경우에는 셋째 줄에는 고장난 버튼이 주어지며, 같은 버튼이 여러 번 주어지는 경우는 없다.

출력

첫째 줄에 채널 N으로 이동하기 위해 버튼을 최소 몇 번 눌러야 하는지를 출력한다.

4. 리모컨

```python
import sys
input = sys.stdin.readline

def possible_num(x):
    x = list(str(x))
    for element in x:
        if element in err:
            return False
    return True

N = int(input())
M = int(input())
err = list(input().strip())

answer = abs(N - 100)

for temp in range(1000001):
    if possible_num(temp) is True:
        answer = min(answer, len(str(temp)) + abs(N-temp))
print(answer)
```

5. DFS와 BFS

DFS와 BFS 성공

☆

실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	203014	74787	44376	35.857%

문제

그래프를 DFS로 탐색한 결과와 BFS로 탐색한 결과를 출력하는 프로그램을 작성하시오. 단, 방문할 수 있는 정점이 여러 개인 경우에는 정점 번호가 작은 것을 먼저 방문하고, 더 이상 방문할 수 있는 점이 없는 경우 종료한다. 정점 번호는 1번부터 N번까지이다.

입력

첫째 줄에 정점의 개수 N(1 ≤ N ≤ 1,000), 간선의 개수 M(1 ≤ M ≤ 10,000), 탐색을 시작할 정점의 번호 V가 주어진다. 다음 M개의 줄에는 간선이 연결하는 두 정점의 번호가 주어진다. 어떤 두 정점 사이에 여러 개의 간선이 있을 수 있다. 입력으로 주어지는 간선은 양방향이다.

출력

첫째 줄에 DFS를 수행한 결과를, 그 다음 줄에는 BFS를 수행한 결과를 출력한다. V부터 방문된 점을 순서대로 출력하면 된다.

5. DFS와 BFS

```python
N, M, V = map(int, input().split())
graph = [[] for _ in range(N+1)]
for _ in range(M):
    start, end = map(int, input().split())
    graph[start].append(end)
    graph[end].append(start)

#DFS
visited = [0] * (N+1)
DFS(graph, visited)
print()
#BFS
visited = [0] * (N+1)
BFS(graph,visited)
```

5. DFS와 BFS

```python
def BFS(graph, visited):

    Q = []
    for element in sorted(graph[V]):
        Q.append(element)
    visited[V] = 1
    print(V, end= " ")
    while Q:
        ne = Q.pop(0)
        if visited[ne] == 0:
            visited[ne] = 1
            print(ne, end = " ")
            for element in sorted(graph[ne]):
                if visited[element] == 0:
                    Q.append(element)
```

```python
def DFS(graph, visited):

    stack = []
    for element in sorted(graph[V], reverse=True):
        stack.append(element)
    visited[V] = 1
    print(V, end=" ")
    while stack:
        ne = stack.pop()
        if visited[ne] == 0:
            visited[ne] = 1
            print(ne, end = " ")
            for element in sorted(graph[ne], reverse=True):
                if visited[element] == 0:
                    stack.append(element)
```

6. 케빈 베이컨의 6단계 법칙

문제

케빈 베이컨의 6단계 법칙에 의하면 지구에 있는 모든 사람들은 최대 6단계 이내에서 서로 아는 사람으로 연결될 수 있다. 케빈 베이컨 게임은 임의의 두 사람이 최소 몇 단계 만에 이어질 수 있는지 계산하는 게임이다.

예를 들면, 전혀 상관없을 것 같은 인하대학교의 이강호와 서강대학교의 민세희는 몇 단계만에 이어질 수 있을까?

천민호는 이강호와 같은 학교에 다니는 사이이다. 천민호와 최백준은 Baekjoon Online Judge를 통해 알게 되었다. 최백준과 김선영은 같이 Startlink를 창업했다. 김선영과 김도현은 같은 학교 동아리 소속이다. 김도현과 민세희는 같은 학교에 다니는 사이로 서로 알고 있다. 즉, 이강호-천민호-최백준-김선영-김도현-민세희 와 같이 5단계만 거치면 된다.

케빈 베이컨은 미국 헐리우드 영화배우들 끼리 케빈 베이컨 게임을 했을때 나오는 단계의 총 합이 가장 적은 사람이라고 한다.

오늘은 Baekjoon Online Judge의 유저 중에서 케빈 베이컨의 수가 가장 작은 사람을 찾으려고 한다. 케빈 베이컨 수는 모든 사람과 케빈 베이컨 게임을 했을 때, 나오는 단계의 합이다.

예를 들어, BOJ의 유저가 5명이고, 1과 3, 1과 4, 2와 3, 3과 4, 4와 5가 친구인 경우를 생각해보자.

1은 2까지 3을 통해 2단계 만에, 3까지 1단계, 4까지 1단계, 5까지 4를 통해서 2단계 만에 알 수 있다. 따라서, 케빈 베이컨의 수는 2+1+1+2 = 6이다.

2는 1까지 3을 통해서 2단계 만에, 3까지 1단계 만에, 4까지 3을 통해서 2단계 만에, 5까지 3과 4를 통해서 3단계 만에 알 수 있다. 따라서, 케빈 베이컨의 수는 2+1+2+3 = 8이다.

3은 1까지 1단계, 2까지 1단계, 4까지 1단계, 5까지 4를 통해 2단계 만에 알 수 있다. 따라서, 케빈 베이컨의 수는 1+1+1+2 = 5이다.

4는 1까지 1단계, 2까지 3을 통해 2단계, 3까지 1단계, 5까지 1단계 만에 알 수 있다. 4의 케빈 베이컨의 수는 1+2+1+1 = 5가 된다.

마지막으로 5는 1까지 4를 통해 2단계, 2까지 4와 3을 통해 3단계, 3까지 4를 통해 2단계, 4까지 1단계 만에 알 수 있다. 5의 케빈 베이컨의 수는 2+3+2+1 = 8이다.

5명의 유저 중에서 케빈 베이컨의 수가 가장 작은 사람은 3과 4이다.

BOJ 유저의 수와 친구 관계가 입력으로 주어졌을 때, 케빈 베이컨의 수가 가장 작은 사람을 구하는 프로그램을 작성하시오.

6. 케빈 베이컨의 6단계 법칙

입력

첫째 줄에 유저의 수 N (2 ≤ N ≤ 100)과 친구 관계의 수 M (1 ≤ M ≤ 5,000)이 주어진다. 둘째 줄부터 M개의 줄에는 친구 관계가 주어진다. 친구 관계는 A와 B로 이루어져 있으며, A와 B가 친구라는 뜻이다. A와 B가 친구이면, B와 A도 친구이며, A와 B가 같은 경우는 없다. 친구 관계는 중복되어 들어올 수도 있으며, 친구가 한 명도 없는 사람은 없다. 또, 모든 사람은 친구 관계로 연결되어져 있다. 사람의 번호는 1부터 N까지이며, 두 사람이 같은 번호를 갖는 경우는 없다.

출력

첫째 줄에 BOJ의 유저 중에서 케빈 베이컨의 수가 가장 작은 사람을 출력한다. 그런 사람이 여러 명일 경우에는 번호가 가장 작은 사람을 출력한다.

예제 입력 1 복사

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

예제 출력 1 복사

```
3
```

6. 케빈 베이컨의 6단계 법칙

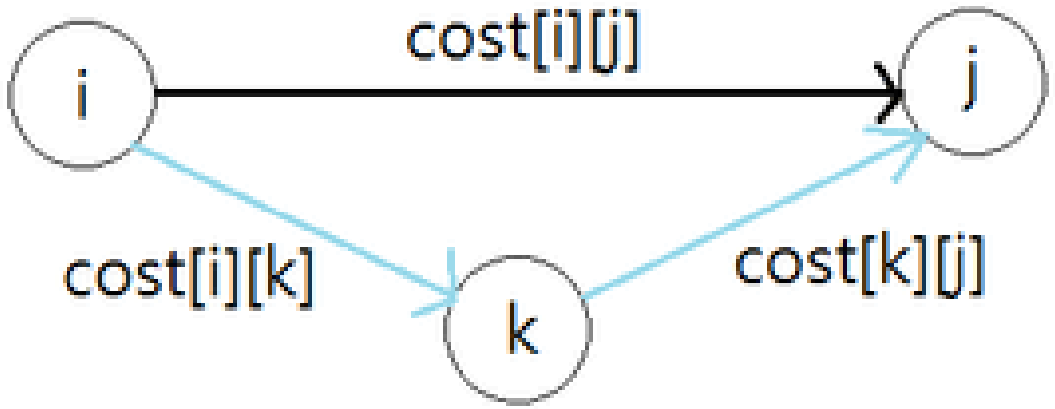
```python
INF = int(1e9)

N, M = map(int, input().split())

relation = [[INF] * N for _ in range(N)]

for i in range(N):
    relation[i][i] = 0

for _ in range(M):
    start, end = map(int, input().split())
    relation[start-1][end-1] = 1
    relation[end-1][start-1] = 1

for k in range(N):
    for a in range(N):
        for b in range(N):
            relation[a][b] = min(relation[a][b], relation[a][k]+relation[k][b])
```

```python
answer = INF
index = INF
for i in range(len(relation)):
    if sum(relation[i]) < answer:
        answer = sum(relation[i])
        index = i
print(index+1)
```

10.1.1 플로이드-워셜(Floyd-Warshall) 알고리즘 [편집]

가장 간단하고 이해하기 쉬운 알고리즘이다. 간단하기 때문에 수행 시간도 생각보다 빠른 편이다. 그래프 상에 비용이 음수인 간선이 존재하는 건 괜찮지만 비용이 음수인 사이클이 존재하지 않아야 한다. 모든 정점 쌍들의 최소 비용을 한번에 구해버린다.

현재까지 구한 i에서 j로 가는 최소 비용보다 i에서 k를 거쳐 j로 가는 최소비용이 더 작다면 $cost\,[i]\,[j]$를 $cost\,[i]\,[k] + cost\,[k]\,[j]$로 대치한다. 이렇게 해서 모든 (i, j)에 대해 $cost\,[i]\,[j]$를 $O\left(n^3\right)$로 구할 수 있다.



7. 1로 만들기

1로 만들기 성공

☆

실버 III

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.15 초 (하단 참고)	128 MB	223243	73379	46907	32.254%

문제

정수 X에 사용할 수 있는 연산은 다음과 같이 세 가지 이다.

1. X가 3으로 나누어 떨어지면, 3으로 나눈다.
2. X가 2로 나누어 떨어지면, 2로 나눈다.
3. 1을 뺀다.

정수 N이 주어졌을 때, 위와 같은 연산 세 개를 적절히 사용해서 1을 만들려고 한다. 연산을 사용하는 횟수의 최솟값을 출력하시오.

입력

첫째 줄에 1보다 크거나 같고, 10^6보다 작거나 같은 정수 N이 주어진다.

출력

첫째 줄에 연산을 하는 횟수의 최솟값을 출력한다.

8. 잃어버린 괄호

잃어버린 괄호 성공

☆

실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	55776	28770	22883	51.182%

문제

세준이는 양수와 +, -, 그리고 괄호를 가지고 식을 만들었다. 그리고 나서 세준이는 괄호를 모두 지웠다.

그리고 나서 세준이는 괄호를 적절히 쳐서 이 식의 값을 최소로 만들려고 한다.

괄호를 적절히 쳐서 이 식의 값을 최소로 만드는 프로그램을 작성하시오.

입력

첫째 줄에 식이 주어진다. 식은 '0'~'9', '+', 그리고 '-'만으로 이루어져 있고, 가장 처음과 마지막 문자는 숫자이다. 그리고 연속해서 두 개 이상의 연산자가 나타나지 않고, 5자리보다 많이 연속되는 숫자는 없다. 수는 0으로 시작할 수 있다. 입력으로 주어지는 식의 길이는 50보다 작거나 같다.

출력

첫째 줄에 정답을 출력한다.

8. 잃어버린 괄호

```python
S = input().split('-')

answer = 0

for element in S[0].split('+'):
    answer += int(element)

for element in S[1:]:
    for sub in element.split('+'):
        answer -= int(sub)

print(answer)
```

```
55-50+40
['55', '50+40']
-35
```

```
1+50-15+25+15-11111-444+1245
['1+50', '15+25+15', '11111', '444+1245']
-12804
```

9. 나는야 포켓몬 마스터 이다솜

나는야 포켓몬 마스터 이다솜 성공

☆

4 실버 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	55337	19299	14433	33.811%

문제

안녕? 내 이름은 이다솜. 나의 꿈은 포켓몬 마스터야. 일단 포켓몬 마스터가 되기 위해선 포켓몬을 한 마리 잡아야겠지? 근처 숲으로 가야겠어.

(뚜벅 뚜벅)

얏! 꼬렛이다. 꼬렛? 귀여운데, 나의 첫 포켓몬으로 딱 어울린데? 내가 잡고 말겠어. 가라! 몬스터볼~

(펑!) 헐랭... 왜 안 잡히지?ㅜㅜ 몬스터 볼만 던지면 되는 게 아닌가...ㅜㅠ

(터벅터벅)

10. 팩토리얼 0의 개수

팩토리얼 0의 개수 성공

☆

실버 V

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	49867	23936	19846	47.920%

문제

N!에서 뒤에서부터 처음 0이 아닌 숫자가 나올 때까지 0의 개수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N이 주어진다. (0 ≤ N ≤ 500)

출력

첫째 줄에 구한 0의 개수를 출력한다.

11. 숨바꼭질

숨바꼭질 성공 다국어

☆ 한국어

실버 I

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	172539	49399	31023	25.164%

문제

수빈이는 동생과 숨바꼭질을 하고 있다. 수빈이는 현재 점 N(0 ≤ N ≤ 100,000)에 있고, 동생은 점 K(0 ≤ K ≤ 100,000)에 있다. 수빈이는 걷거나 순간이동을 할 수 있다. 만약, 수빈이의 위치가 X일 때 걷는다면 1초 후에 X-1 또는 X+1로 이동하게 된다. 순간이동을 하는 경우에는 1초 후에 2*X의 위치로 이동하게 된다.

수빈이와 동생의 위치가 주어졌을 때, 수빈이가 동생을 찾을 수 있는 가장 빠른 시간이 몇 초 후인지 구하는 프로그램을 작성하시오.

입력

첫 번째 줄에 수빈이가 있는 위치 N과 동생이 있는 위치 K가 주어진다. N과 K는 정수이다.

출력

수빈이가 동생을 찾는 가장 빠른 시간을 출력한다.

예제 입력 1 복사

```
5 17
```

예제 출력 1 복사

```
4
```

11. 숨바꼭질

```python
from collections import deque
L = int(1e9)
def search(arr, N, K):
    Q = deque()
    Q.append(N)

    while Q:
        x = Q.popleft()
        if x == K:
            return arr[x]

        for j in (x + 1, x - 1, x * 2):
            if (0 <= j < L) and arr[j] == 0:
                arr[j] = arr[x] + 1
                Q.append(j)

N, K = map(int, input().split())
arr = [0] * L
answer = search(arr, N, K)
print(answer)
```

12. 듣보잡

듣보잡

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	61181	25576	19530	40.311%

문제

김진영이 듣도 못한 사람의 명단과, 보도 못한 사람의 명단이 주어질 때, 듣도 보도 못한 사람의 명단을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 듣도 못한 사람의 수 N, 보도 못한 사람의 수 M이 주어진다. 이어서 둘째 줄부터 N개의 줄에 걸쳐 듣도 못한 사람의 이름과, N+2째 줄부터 보도 못한 사람의 이름이 순서대로 주어진다. 이름은 띄어쓰기 없이 알파벳 소문자로만 이루어지며, 그 길이는 20 이하이다. N, M은 500,000 이하의 자연수이다.

듣도 못한 사람의 명단에는 중복되는 이름이 없으며, 보도 못한 사람의 명단도 마찬가지이다.

출력

듣보잡의 수와 그 명단을 사전순으로 출력한다.

예제 입력 1 복사

```
3 4
ohhenrie
charlie
baesangwook
obama
```

예제 출력 1 복사

```
2
baesangwook
ohhenrie
```

13. 종이의 개수

종이의 개수 성공

☆

실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	32734	19317	14541	58.395%

문제

N×N크기의 행렬로 표현되는 종이가 있다. 종이의 각 칸에는 -1, 0, 1 중 하나가 저장되어 있다. 우리는 이 행렬을 다음과 같은 규칙에 따라 적절한 크기로 자르려고 한다.

1. 만약 종이가 모두 같은 수로 되어 있다면 이 종이를 그대로 사용한다.
2. (1)이 아닌 경우에는 종이를 같은 크기의 종이 9개로 자르고, 각각의 잘린 종이에 대해서 (1)의 과정을 반복한다.

이와 같이 종이를 잘랐을 때, -1로만 채워진 종이의 개수, 0으로만 채워진 종이의 개수, 1로만 채워진 종이의 개수를 구해내는 프로그램을 작성하시오.

입력

첫째 줄에 N($1 \le N \le 3^7$, N은 3^k 꼴)이 주어진다. 다음 N개의 줄에는 N개의 정수로 행렬이 주어진다.

출력

첫째 줄에 -1로만 채워진 종이의 개수를, 둘째 줄에 0으로만 채워진 종이의 개수를, 셋째 줄에 1로만 채워진 종이의 개수를 출력한다.

14. 최소 힙

최소 힙 성공

실버 II

시간 제한	메모리 제한	제출	정답	맞힌
1 초 (추가 시간 없음) (하단 참고)	128 MB	52082	24227	190

문제

널리 잘 알려진 자료구조 중 최소 힙이 있다. 최소 힙을 이용하여 다음과 같은 연산을 지원하는 프로그램을 작성하시오.

1. 배열에 자연수 x를 넣는다.
2. 배열에서 가장 작은 값을 출력하고, 그 값을 배열에서 제거한다.

프로그램은 처음에 비어있는 배열에서 시작하게 된다.

입력

첫째 줄에 연산의 개수 N(1 ≤ N ≤ 100,000)이 주어진다. 다음 N개의 줄에는 연산에 대한 정보를 나타내는 정수 x가 주어
가하는) 연산이고, x가 0이라면 배열에서 가장 작은 값을 출력하고 그 값을 배열에서 제거하는 경우이다. x는 2^{31}보다 작은
않는다.

출력

입력에서 0이 주어진 횟수만큼 답을 출력한다. 만약 배열이 비어 있는 경우인데 가장 작은 값을 출력하라고 한 경우에는 0

```python
import sys
import heapq
input = sys.stdin.readline

N = int(input())
Q = []
for _ in range(N):
    temp = int(input())

    if temp > 0 :
        heapq.heappush(Q, temp)
    elif temp == 0:
        if len(Q) != 0:
            print(heapq.heappop(Q))
        elif len(Q) == 0:
            print(0)
```

15. 회의실 배정

문제

한 개의 회의실이 있는데 이를 사용하고자 하는 N개의 회의에 대하여 회의실 사용표를 만들려고 한다. 각 회의 I에 대해 시작시간과 끝나는 시간이 주어져 있고, 각 회의가 겹치지 않게 하면서 회의실을 사용할 수 있는 회의의 최대 개수를 찾아보자. 단, 회의는 한번 시작하면 중간에 중단될 수 없으며 한 회의가 끝나는 것과 동시에 다음 회의가 시작될 수 있다. 회의의 시작시간과 끝나는 시간이 같을 수도 있다. 이 경우에는 시작하자마자 끝나는 것으로 생각하면 된다.

입력

첫째 줄에 회의의 수 N(1 ≤ N ≤ 100,000)이 주어진다. 둘째 줄부터 N+1 줄까지 각 회의의 정보가 주어지는데 이것은 공백을 사이에 두고 회의의 시작시간과 끝나는 시간이 주어진다. 시작 시간과 끝나는 시간은 2^{31}-1보다 작거나 같은 자연수 또는 0이다.

출력

첫째 줄에 최대 사용할 수 있는 회의의 최대 개수를 출력한다.

예제 입력 1 복사

```
11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14
```

예제 출력 1 복사

```
4
```

15. 회의실 배정

```python
import sys

N = int(sys.stdin.readline())
meeting = sorted([list(map(int, sys.stdin.readline().split())) for i in range(N)], key=lambda x: (x[1], x[0]))

end = answer = 0
for m in meeting:
    if end <= m[0]:
        end = m[1]
        answer += 1
print(answer)
```

교수님은 기다리지 않는다!

교수님은 기다리지 않는다 다국어

☆ 한국어

플래티넘 III

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	7154	1926	1249	26.119%

문제

상근이는 매일 아침 실험실로 출근해서 샘플의 무게를 재는 일을 하고 있다. 상근이는 두 샘플을 고른 뒤, 저울을 이용해서 무게의 차이를 잰다.

교수님의 마음에 들기 위해서 매일 아침부터 무게를 재고 있지만, 가끔 교수님이 실험실에 들어와서 상근이에게 어떤 두 샘플의 무게의 차이를 물어보기도 한다. 이때, 상근이는 지금까지 잰 결과를 바탕으로 대답을 할 수도 있고, 못 할 수도 있다.

상근이는 결과를 출근한 첫 날부터 공책에 적어 두었다. 하지만, 엄청난 양의 무게가 적혀있기 때문에, 교수님의 질문에 재빨리 대답할 수가 없었다. 이런 상근이를 위해서 프로그램을 만들려고 한다.

상근이가 실험실에서 한 일이 순서대로 주어진다. 어떤 두 샘플의 무게의 차이를 구할 수 있는지 없는지를 알아내는 프로그램을 작성하시오.

입력

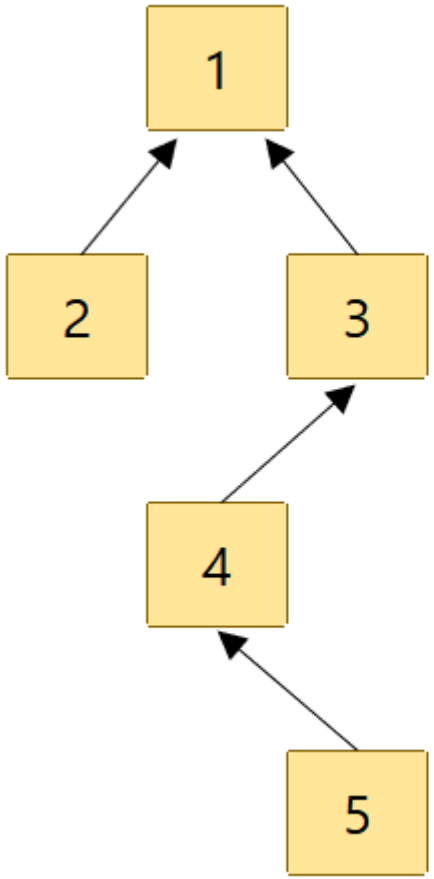
입력은 여러 개의 테스트 케이스로 이루어져 있다.

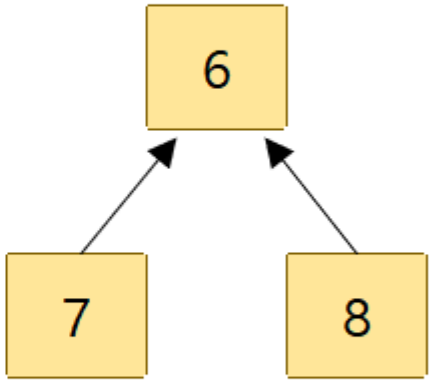
각 테스트 케이스의 첫째 줄에는 샘플의 종류의 개수 N ($2 \le N \le 100,000$)과 상근이가 실험실에서 한 일의 수 M ($1 \le M \le 100,000$)이 주어진다. 샘플은 1번부터 N번까지 번호가 매겨져 있다. 다음 M개 줄에는 상근이가 실험실에서 한 일이 주어진다.

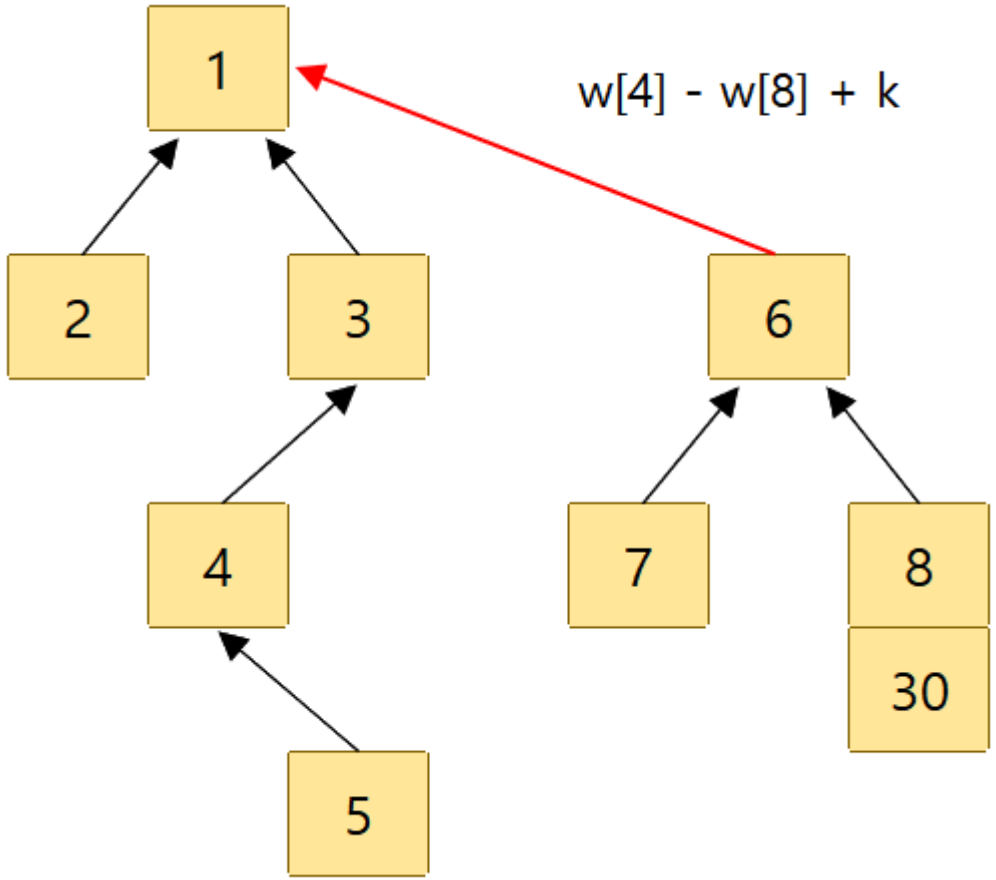
상근이가 무게를 쟀다면, ! a b w와 같은 형식으로 주어진다. 이 뜻은 b가 a보다 w그램 무겁다는 뜻이다. ($a \neq b$) w는 1,000,000을 넘지 않는 음이 아닌 정수이다. 모든 측정은 정확하고, 일관성을 유지한다.

교수님의 질문은 ? a b와 같은 형식으로 주어진다. 이 뜻은 b가 a보다 얼마나 무거운지를 출력하라는 뜻이다.

마지막 줄에는 0이 두 개 주어진다.


1
2
3
4
5


6
7
8


1
w[4] - w[8] + k
2
3
6
4
7
8
30
5

```python
while True:
    n, m = map(int, input().split())

    if n == 0 and m == 0: #종료 조건
        break

    parent = [i for i in range(n + 1)] #부모 테이블
    diff = [0 for i in range(n + 1)] #얼마의 차이가 있는지 나타내는 테이블

    for _ in range(m):
        input_list = list(map(str, input().split()))
        front, rear = int(input_list[1]), int(input_list[2])
        _, _ = find(front), find(rear) #테이블 업데이트
        if input_list[0] == "!":
            union(front, rear, int(input_list[3])) #무게는 있을 수도 없을 수도 있음
        else:
            if parent[front] == parent[rear]:
                print(diff[rear] - diff[front])
            else:
                print("UNKNOWN")
```

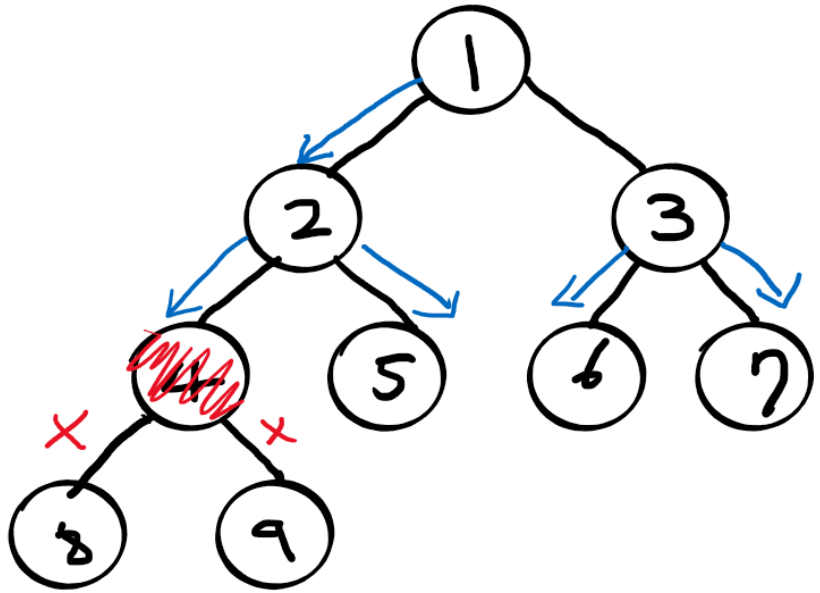
```python
import sys
input = sys.stdin.readline
sys.setrecursionlimit(10**6)

def find(x):
    if x == parent[x]:
        return x
    else:
        root = find(parent[x]) #부모가 있는경우
        diff[x] += diff[parent[x]] #거리 업데이트
        parent[x] = root #부모 지정
        return parent[x]

def union(x, y, k):
    x_root = parent[x]
    y_root = parent[y]
    if x_root != y_root: #부모가 같지 않는경우 == 비교한적 없는 경우
        parent[y_root] = x_root
        diff[y_root] = (diff[x] + k) - diff[y]
```

What is Back Tracking

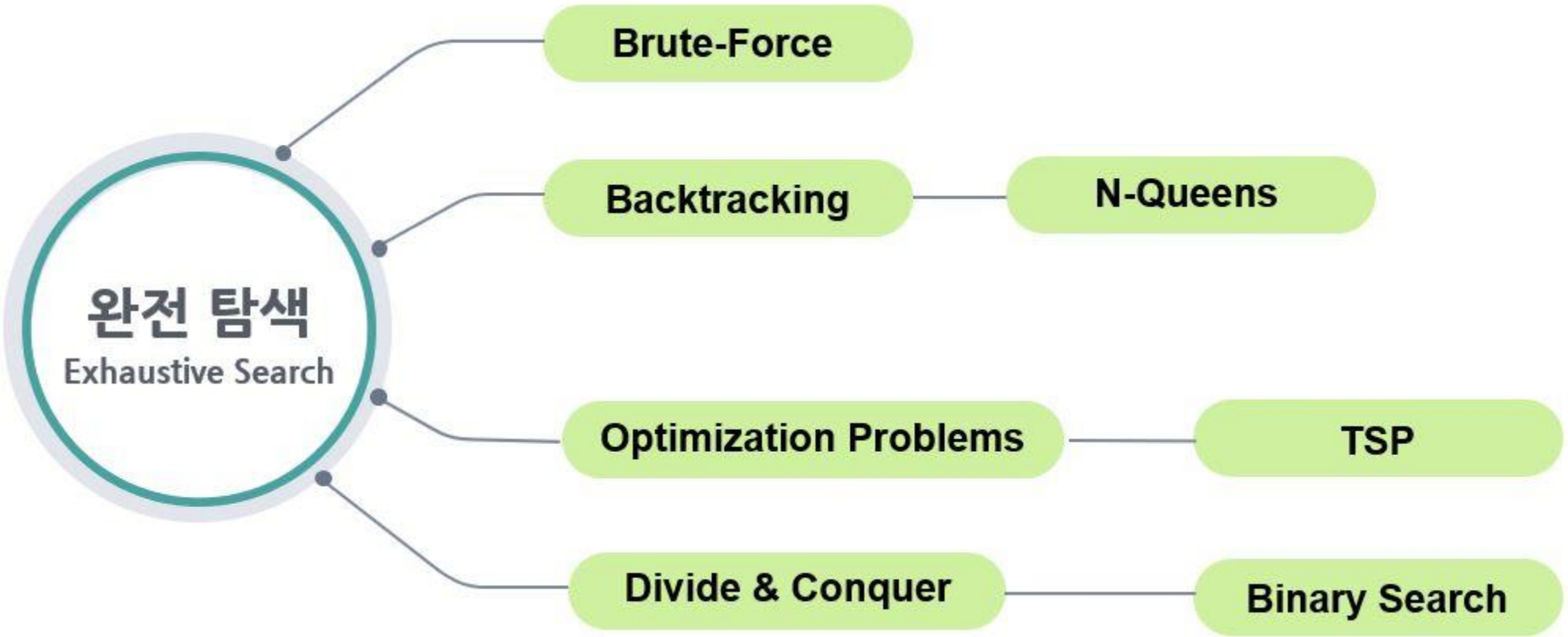
모든 경우의 수를 검사 하던 도중 → 정답이 될 수 없는 방향으로는 가지 않는 것!!!

특별한 방법이 아니라, 논리적으로 정답이 될 수 없는 것은 더 이상 진행하지 않는 방식

주로 탐색에서 사용 됨



Back Tracking



N-Queen 성공

☆

골드 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
10 초	128 MB	76422	36977	24163	47.331%

문제

N-Queen 문제는 크기가 N × N인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제이다.

N이 주어졌을 때, 퀸을 놓는 방법의 수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N이 주어진다. (1 ≤ N < 15)

출력

첫째 줄에 퀸 N개를 서로 공격할 수 없게 놓는 경우의 수를 출력한다.

예제 입력 1 복사

```
8
```

예제 출력 1 복사

```
92
```

Back Tracking

```python
n = int(input())
s = [0 for i in range(16)]
result = 0
def isTrue(x):
    for i in range(1, x):
        if s[x] == s[i] or abs(s[x] - s[i]) == x - i:
            return False
    return True
def dfs(cnt):
    global result
    if cnt > n:
        result += 1
    else:
        for i in range(1, n + 1):
            s[cnt] = i
            if isTrue(cnt):
                dfs(cnt + 1)
dfs(1)
print(result)
```

그 순간 가장 좋은 선택을 하는 것!!!

→ 참고 자료 활용

코딩테스트 리뷰

문제 풀이 상황 : 1,2번 풀고 -> 3번 정확성 -> 4번 ->다시 3번

Point : 잘 풀었던 건 굳이 묻지 않음, 어려워 하거나, 못푼 것에 대한 질문

자료구조

Q1. 멀티쓰레딩 → 여러 쓰레드가 같은 자원을 공유한다. 어떤 문제와 해결책이 있는가

Q2. 뮤텍스와 세마포어의 차이점

Q3. 주 사용 언어는 ? Thread safe하다는 말 들어봤냐?

Q4. Thread Safe하게 구성할려고 하면 어떻게 하면 될까요 ?

Q5. 순서 관계를 지킨다는 말은 ?

Q6. 리스트, 셋, 맵, (딕셔너리) 각각의 특징을 설명해 달라

Q7. 스택, 큐는 아시죠 ?

Q8. Hash Table을 구성할려고 한다. 어떤 부분을 고려해야할까 ?

Q9. 충돌 시 어떻게 다른 곳에 저장할 것인가?

자료구조

Q10. 재귀함수로 코딩을 하게 되는데, 어떤 점들을 고려해서 써야하는가 ?

Q11. 잘못 짜면 어떻게 될 수 있는가 ?

Q12. LRU cache를 구현할려고 한다. 어떻게 할 수 있을까?

Q13. 그렇다면 어떤 자료형으로 LRU Cache를 구성할 수 있을까?

→ https://dailylifeofdeveloper.tistory.com/355

Q14. cache hit를 못하면 어쩔까요 ?

Q15. 뭘 위해서 list를 이용하냐 → 이건 힌트!

Q16. 서비스를 만들거다. 최근 30간 이벤트 발생횟수를 return하는 서비스를 어떤 자료구조로 만들것인가? Input(이벤트수몇개야), return (30초가 이벤트 발생횟수)

Q17. return 값을 얻는데까지의 시간 복잡도는 ?

QnA