BOJ

#7576. 토마토

https://www.acmicpc.net/problem/7576

25.05.19

📝 Problem 시간: 1시간 15분

#7576. 토마토

토마토

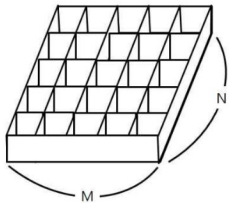
☆

골드 V

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	218795	88595	56363	37.848%

문제

철수의 토마토 농장에서는 토마토를 보관하는 큰 창고를 가지고 있다. 토마토는 아래의 그림과 같이 격자 모양 상자의 칸에 하나씩 넣어서 창고에 보관한다.



창고에 보관되는 토마토들 중에는 잘 익은 것도 있지만, 아직 익지 않은 토마토들도 있을 수 있다. 보관 후 하루가 지나면, 익은 토마토들의 인접한 곳에 있는 익지 않은 토마토들은 익은 토마토의 영향을 받아 익게 된다. 하나의 토마토의 인접한 곳은 왼쪽, 오른쪽, 앞, 뒤 네 방향에 있는 토마토를 의미한다. 대각선 방향에 있는 토마토들에게는 영향을 주지 못하며, 토마토가 혼자 저절로 익는 경우는 없다고 가정한다. 철수는 창고에 보관된 토마토들이 며칠이 지나면 다 익게 되는지, 그 최소 일수를 알고 싶어 한다.

토마토를 창고에 보관하는 격자모양의 상자들의 크기와 익은 토마토들과 익지 않은 토마토들의 정보가 주어졌을 때, 며칠이 지나면 토마토들이 모두 익는지, 그 최소 일수를 구하는 프로그램을 작성하라. 단, 상자의 일부 칸에는 토마토가 들어있지 않을 수도 있다.

💡 Hint

Step 1. 문제 분석

💡 문제 요약

토마토 창고가 2차원 격자 형태로 주어지고, 각 칸에는 다음 중 하나가 들어있다.

- **1**: 익은 토마토
- **0**: 익지 않은 토마토
- **-1**: 토마토 없음 (빈 칸)

익은 토마토는 하루가 지나면 **상하좌우** 인접한 토마토를 익게 만든다.

며칠이 지나면 **모든 토마토가 익을 수 있는지**, **최소 며칠이 걸리는지** 구해야 한다.

- 이미 모두 익어있으면 **0** 출력
- 도저히 모두 익지 못하는 경우 **1** 출력

💡 Hint

Step 1. 문제 분석

💡 문제 유형

- 이 문제는 전형적인 **그래프 탐색 문제**이다.
- **“최소 일수”, “왼쪽, 오른쪽, 앞, 뒤 네 방향”, “주변의 토마토들을 익힌다”**는 내용이 들어가 있으므로 **“너비 우선 탐색(BFS)”** 문제 유형이라는 것을 알 수 있다.
- 거리(=일수)를 측정해야 하므로 BFS의 **레벨 탐색** 특성을 그대로 활용하면 된다

💡 Hint

Step 2. 접근 방식

💡 풀이 아이디어

1. 모든 익은 토마토(`1`)를 BFS의 **초기 노드**로 큐에 삽입
2. BFS를 돌리며, 인접한 익지 않은 토마토(`0`)가 있는 위치에 1씩 더해준다.
   (날짜 역할로써 여기서 나온 제일 큰 값이 정답이 된다.)
3. BFS가 종료된 후,
   - 아직 0인 토마토가 있다면 `-1` 출력
   - 그 외에는 익은 값들 중 최대값에서 `1`을 빼 출력
     - 1을 빼는 이유는 처음에 1(익힌 토마토)부터 시작했기 때문

Hint

Step 3. 코드 플로우

1. 입력 처리

- 첫 줄에서 창고의 가로 `M`, 세로 `N` 크기를 입력받는다.
- 그 다음 `N`줄에 걸쳐 토마토 상태가 담긴 2차원 리스트 `box`를 생성한다.
  - `1`: 익은 토마토
  - `0`: 익지 않은 토마토
  - `-1`: 토마토가 없는 칸

2. 초기 설정

- 큐(`deque`)를 생성하고, 익은 토마토(`1`)의 좌표를 모두 큐에 넣는다.
- 방향 이동을 위한 `dx`, `dy` 벡터(상, 우, 하, 좌)를 설정한다.
- 정답 변수 `day = 0`을 선언해, 최종 일수를 저장할 준비를 한다.

3. BFS 수행

- 큐가 빌 때까지 다음을 반복한다:
  - 큐에서 `(x, y)`를 꺼낸다.
  - 네 방향으로 이동하면서 인접한 칸이 익지 않은 토마토(`0`)라면:
    - 해당 칸의 값을 `현재 값 + 1`로 갱신한다 (며칠 째 익은 건지 누적)
    - 그 칸을 큐에 추가하여 다음 차례에 탐색할 수 있도록 한다.

4. 결과 계산

- 탐색이 끝난 후 2중 반복문으로 box 전체를 확인한다:
  - 0이 남아 있다면 익지 못한 토마토가 있다는 뜻이므로 -1을 출력하고 종료한다.
  - 그렇지 않다면 가장 큰 값을 찾아 day에 저장한다.

5. 정답 출력

- 1부터 시작했기 때문에 정답은 day - 1이다. 이를 출력한다.

✨Solution

💡 BFS 풀이

대체로 풀이 아이디어가 비슷하다

- [백준] 7576: 토마토(파이썬/해설포함)
- [백준] 7576번 - 토마토 | 파이썬

```python
import sys
from collections import deque
input = sys.stdin.readline

# 1. 입력 처리
M, N = map(int, input().split())                          # 가로 칸 수, 세로 칸 수
box = [list(map(int, input().split())) for _ in range(N)]   # 토마토

# 2. 초기 설정
queue = deque([])                                 # 큐
directions = [(-1, 0), (0, 1), (1, 0), (0, -1)] # 방향벡터
day = 0                                           # 정답이 담길 변수

# 3. 큐에 초기 익은 토마토 위치 저장
for i in range(N):
    for j in range(M):
        if box[i][j] == 1:
            queue.append((i, j))
```

```python
# 4. BFS 탐색
while queue:
    # 처음 토마토 꺼내기
    x, y = queue.popleft()

    # 처음 토마토의 인접한 토마토 찾기
    for dx, dy in directions:
        nx, ny = x + dx, y + dy

        # 범위 내에 있고, 토마토가 익지 않은 경우
        if (0 <= nx < N and 0 <= ny < M) and (box[nx][ny] == 0):
            # 익히고 1 더해주며 횟수 세기
            # 여기서 나온 제일 큰 값이 정답이 된다.
            box[nx][ny] += box[x][y] + 1  # 일수 누적
            queue.append((nx, ny))


# 5. 정답 구하기
for row in box:
    for tomato in row:
        # 모두 탐색했지만 토마토가 모두 익지 않았다면 -1 출력
        if tomato == 0:
            print(-1)
            exit()

    # 다 익혔다면 최댓값이 정답
    day = max(day, max(row))

# 6. 정답 출력
print(day - 1) # 처음에 1로 익은 토마토를 표현했으니 1을 빼준다.
```

💻 Assignment

백준 #7569. 토마토(골드5)

토마토 3차원 문제 (BFS)

- 발제 문제와 제목이 똑같지만 다른 문제입니다. (발제 문제: #7576. 과제 문제: #7569)
- 발제 문제가 2차원이라면, 과제 문제는 3차원입니다.

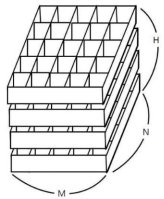
토마토 성공

골드 V

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	109323	48337	35389	43.369%

문제

철수의 토마토 농장에서는 토마토를 보관하는 큰 창고를 가지고 있다. 토마토는 아래의 그림과 같이 격자모양 상자의 칸에 하나씩 넣은 다음, 상자들을 수직으로 쌓아 올려서 창고에 보관한다.



창고에 보관되는 토마토들 중에는 잘 익은 것도 있지만, 아직 익지 않은 토마토들도 있을 수 있다. 보관 후 하루가 지나면, 익은 토마토들의 인접한 곳에 있는 익지 않은 토마토들은 익은 토마토의 영향을 받아 익게 된다. 하나의 토마토에 인접한 곳은 위, 아래, 왼쪽, 오른쪽, 앞, 뒤 여섯 방향에 있는 토마토를 의미한다. 대각선 방향에 있는 토마토들에게는 영향을 주지 못하며, 토마토가 혼자 저절로 익는 경우는 없다고 가정한다. 철수는 창고에 보관된 토마토들이 며칠이 지나면 다 익게 되는지 그 최소 일수를 알고 싶어 한다.

토마토를 창고에 보관하는 격자모양의 상자들의 크기와 익은 토마토들과 익지 않은 토마토들의 정보가 주어졌을 때, 며칠이 지나면 토마토들이 모두 익는지, 그 최소 일수를 구하는 프로그램을 작성하라. 단, 상자의 일부 칸에는 토마토가 들어있지 않을 수도 있다.