2주차 _ 스택 / 큐 알고리즘 풀이

Created	@August 12, 2021
Reviewed	☑
Type	Study Group
기한	@August 15, 2021

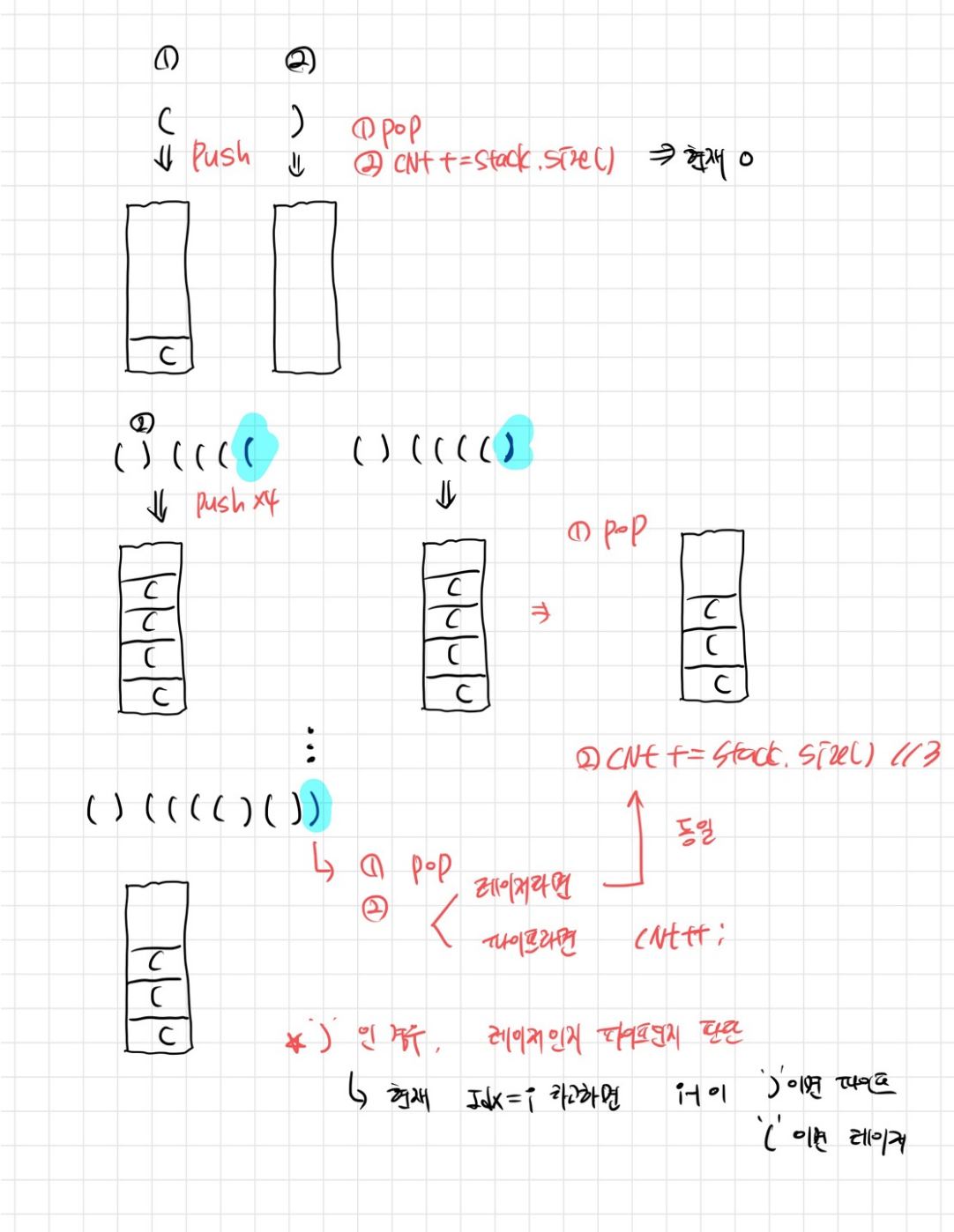
스택

1. 쇠막대기

> 10799번: 쇠막대기
> 여러 개의 쇠막대기를 레이저로 절단하려고 한다. 효율적인 작업을 위해서 쇠막대기를 아래에서 위로 겹쳐 놓고, 레이저를 위에서 수직으로 발사하여 쇠막대기들을 자른다. 쇠막대기와 레이저의 배치는 다음 조건을 만
> https://www.acmicpc.net/problem/10799

▼ 접근방법

- (: 스택에 push
-) : 스택 값을 pop하고, ')' 레이저인지, 파이프인지 판단해줘야함.
  - 현 위치:i 그 이전 위치: i-1 모두가) 인 경우 (즉 ')) ' 이라면) 현재 i는 파이프이다. 파이프 끝이므로, cnt+1해줘야한다.
  - 현 위치:i 그 이전 위치: i-1 이 다른 경우, (즉 ' () ' 이라면) 현재 i는 레이저이다. 현재 스택에 쌓여진 사이즈만큼 cnt+=stack.size()해주어야함.




```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.Stack;

public class Baekjoon_10799 {
```

```
  public static void main(String[] args) throws IOException {
    // TODO Auto-generated method stub

    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    Stack<Character> stack = new Stack<>();

    String str = br.readLine();
    char[] arr = str.toCharArray();

    int cnt = 0;
    stack.push(arr[0]);
    for (int i = 1; i < arr.length; i++) {
      char current = arr[i];

      if (current == '(')
        stack.push(current);

      if (current == ')') { // )인 경우, 그 앞전의 값을 비교하여 지금 )이 파이프인지 레이저인지 판단해야함
        char before = arr[i - 1];
        stack.pop();

        if (before == ')') { // 그 앞전에 레이저였음. 지금은 파이프가 닫힌것
          cnt++; // 막대기 하나가 끝나서 끝 부분만 남은 상태이므로 +1 필수
        }
        else { // 그 앞 (이었으며 현재 )인 상태
          cnt += stack.size(); // 스택사이즈만큼 현재 쇠막대기 수를 카운트 해줌
        }
      }
    }
    System.out.println(cnt);
  }
}
```

2. 창고 다각형

2304번: 창고 다각형

N 개의 막대 기둥이 일렬로 세워져 있다. 기둥들의 폭은 모두 1 m이며 높이는 다를 수 있다. 이 기둥들을 이용하여 양철로 된 창고를 제작하려고 한다. 창고에는 모든 기둥이 들어간다. 이 창고의 지붕을 다음과 같이

https://www.acmicpc.net/problem/2304

BAEKJOON ONLINE JUDGE

큐

1. [백준] 15828 - 라우터

15828번: Router

인터넷을 사용하기 위해서는 컴퓨터에 인터넷 회선을 연결하거나 Wi-Fi를 연결해야 한다. 이렇게 연결된 네트워크를 통해 컴퓨터에는 통신이 가능하다. 마음에 드는 노래나 동영상이 있는 곳에 파일을 전송해달라는 요

https://www.acmicpc.net/problem/15828

▼ 접근방법

- 0 이 입력 : ans 큐에서 값 제거
- -1이 입력: 현재 ans큐 출력 (ans 큐가 비었다면 empty, 값이 있다면 들어온 순서로 출력)
- 그 외의 입력: ans 큐에 값 넣기 (단 현재 버퍼사이즈N을 넘어선다면 넣으면 안됨.)

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

public class Baekjoon_15828 {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    Queue<Integer> ansQ = new LinkedList<Integer>();
    Queue<Integer> cmdQ = new LinkedList<Integer>();

    int N = sc.nextInt(); //버퍼사이즈
    int userInput = 0;

//입력->배열로 처리해도 무방
    do {
      userInput = sc.nextInt();
      cmdQ.add(userInput);
    } while (userInput != -1);

//입력 받은 문장을 하나씩 처리
    while (!cmdQ.isEmpty()) {
      int val = cmdQ.poll();

      if(val == -1) // 명령문이 끝남
        break;
      if (val == 0) { //ans 큐에서 값 하나 빼기
        ansQ.poll();
      }
      else { // 그외의 수는 ans큐에 다 넣기
        if (ansQ.size() < N) {
```

```
          ansQ.add(val);
        }
      }
    }
// while문이 종료되고 ans 큐상태에 따라 출력이 달라진다.
    if (ansQ.isEmpty()) {
      System.out.println("empty");
    }
    else {
      while (!ansQ.isEmpty()) {
        System.out.print(ansQ.poll()+" ");
      }
    }
  }
}
```

2. [백준] 3190 뱀 - 라우터

3190번: 뱀

'Dummy' 라는 도스게임이 있다. 이 게임에는 뱀이 나와서 기어다니는데, 사과를 먹으면 뱀 길이가 늘어난다. 뱀이 이리저리 기어다니다가 벽 또는 자기자신의 몸과 부딪히면 게임이 끝난다.

https://www.acmicpc.net/problem/3190

BAEKJOON ONLINE JUDGE

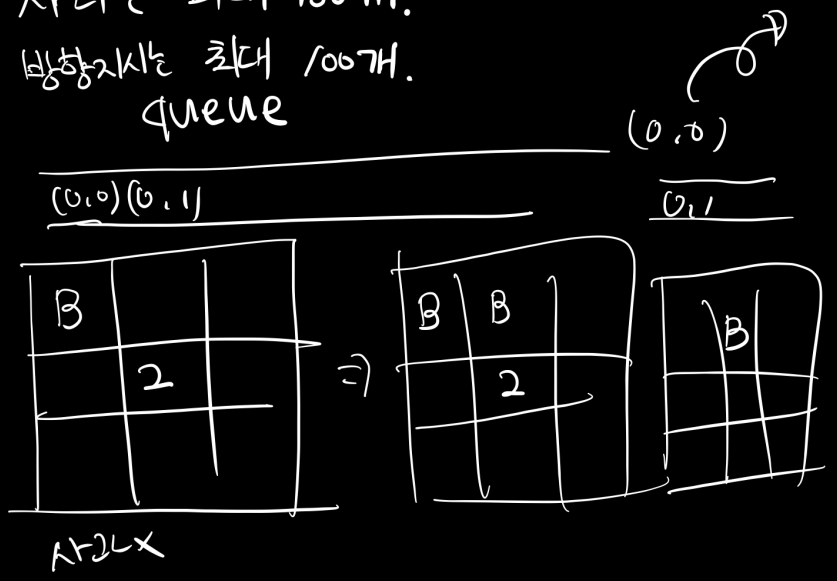
▼ 접근방법

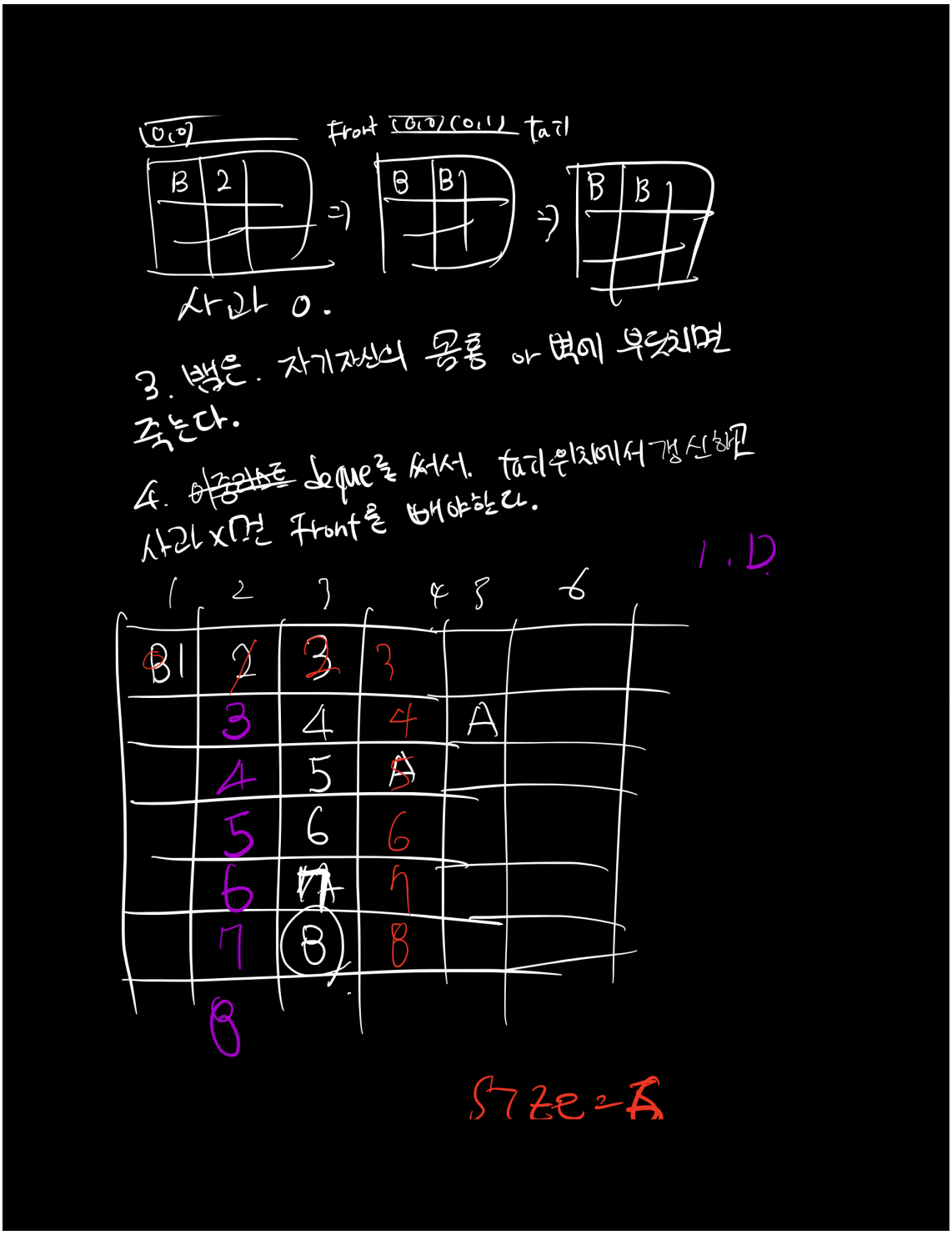
3190 뱀!!!

① 뱀은 (0,0)에위치하고
초당 우측으로 한칸씩이동.

② 이동시 (0,0)(0,1) 2칸을 차지하고
사과가 없으면 (0,0)칸을 비운다.
(큐에 저장해야함)

- 보드의크기는 최소 4×4 최대 100×100
- 사과는 최대 100개.
- 방향전환은 최대 100개.





빨간색 숫자로 이동시켜야함,,

```
import java.util.LinkedList;
import java.util.Scanner;
```

```
public class Main {
  static LinkedList<Point> snake = new LinkedList<>();
  static Point[] apples;
  static Point[] dirs;
  static int N;
  static int K;
  static int L;
  static int dx[] = {0,1,0,-1};
  static int dy[] = {1,0,-1,0};
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    //보드의 크기 N, 사과의 갯수 K, 방향변환 횟수 L
    N = sc.nextInt();
    K = sc.nextInt();
    apples = new Point[K];
    for(int i = 0; i < K; i++) {
      apples[i] = new Point(sc.nextInt(),sc.nextInt());
    }
    L = sc.nextInt();
    dirs = new Point[L];
    //방향변환 인풋
    for(int i = 0; i < L; i++) {
      dirs[i] = new Point(sc.nextInt(), sc.next());
    }
    move();
//    System.out.println(dirs[0].y);
//    int a = 'D';
//    System.out.println(a);
  }

  static void move() {
    //시간초
    int count = 0;
    int index = 0;
    //0 우 1 하 2 좌 3 상
    int dir = 0;
    //뱀의 초기위치
    snake.offer(new Point(0,0));
    while(true) {

      //방향전환
      if( index < L && dirs[index].x == count ) {
        dir = Math.abs(calcdir(dirs[index].y) + dir) % 4;
        index++;
      }


      int nx = dx[dir] + snake.getLast().x;
      int ny = dy[dir] + snake.getLast().y;
      ++count;
      //벽에 박았을때
      if(nx < 0 || nx >= N || ny <0 || ny >= N ) {
        break;
      }
      //자기 꼬리에 박았을때
```

```
      //linkedlist를 마지막 인덱스를 제외하고 순회해야함..
      int flag = 0;
//      System.out.println(snake.size());
      for(int i = 0; i < snake.size(); i++) {
//        System.out.printf("count : %d x : "
//            + "%d nx %d y %d ny %d%n",count,snake.get(i).x,nx,snake.get(i).y,ny);
        if (snake.get(i).x == nx && snake.get(i).y == ny ) {
//          System.out.println("break;");//break문 밖으로 빼야함
          flag = 1;
          break;
        }
      }
      if(flag == 1) break;
      //이제 큐에 넣고 사과의 유무에따라 팝의 유무가 결정됨
      snake.offer(new Point(nx, ny));
      //사과가 있는지 없는지 확인 있으면 컨티뉴 없으면 큐의 head를 팝
      for(Point a : apples) {
//        System.out.printf("count : %d x : "
//            + "%d nx %d y %d ny %d%n",count,a.x,nx+1,a.y,ny+1);
        if (a.x == nx + 1 && a.y == ny + 1 ) {
//          System.out.println("continue");
          a.x =-1;
          a.y =-1;
          flag = 1;
          break;
        }
      }
      if(flag == 1) continue;
        snake.poll();
//      for(Point a : snake) {
//        System.out.printf(a.x + "," + a.y + " ");
//      }



    }//end while
    System.out.println(count);
  }

  static int calcdir(int dir) {
    return dir == 'D' ? 1 : 3;
  }

  static class Point{
    int x, y;
    public Point(int x, int y) {
      // TODO Auto-generated constructor stub
      this.x = x;
      this.y = y;
    }
    public Point(int x, String y) {
      // TODO Auto-generated constructor stub
      this.x = x;
      this.y = y.charAt(0);
    }
```

```
    }

}
```