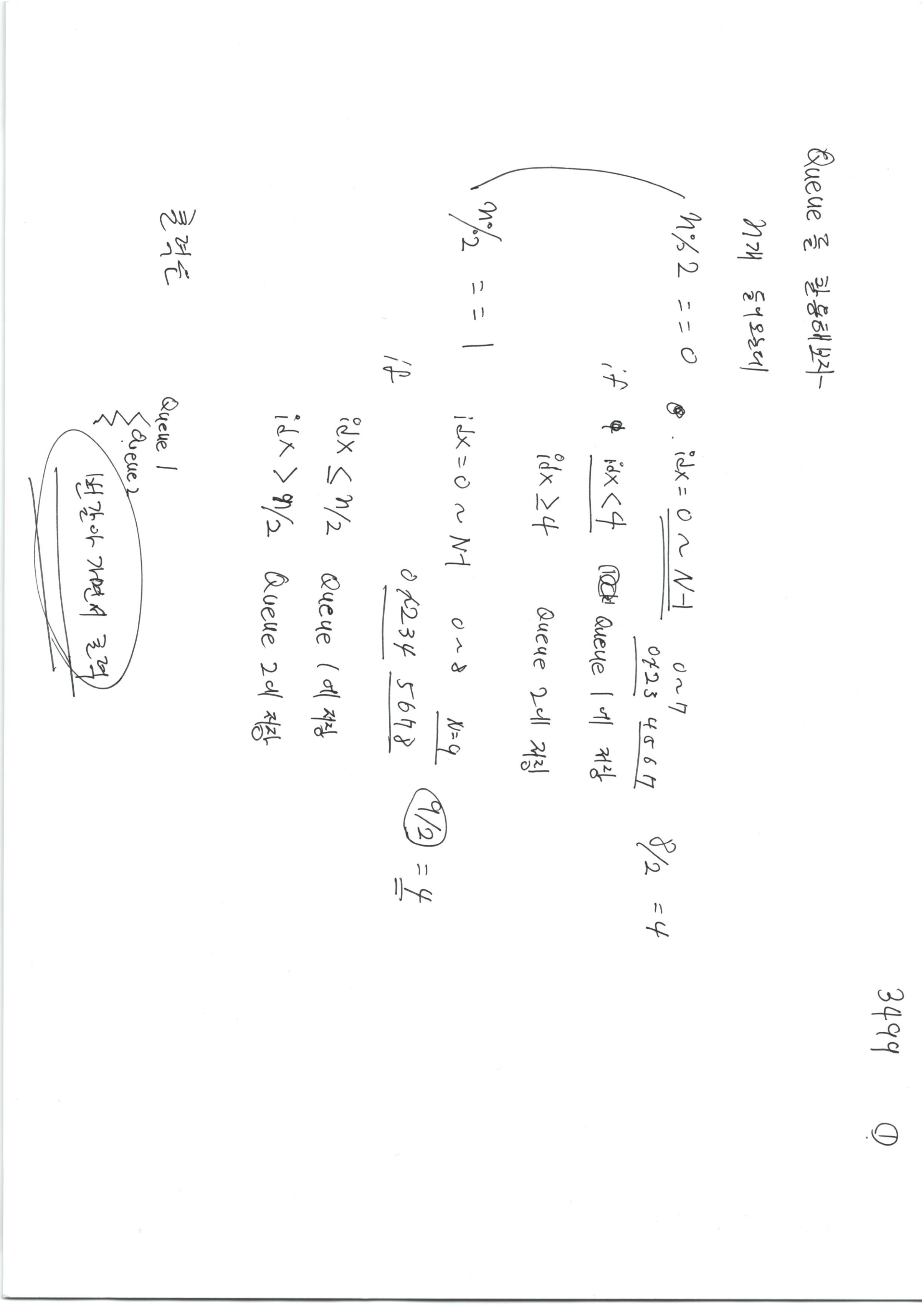
SWEA 3499 퍼펙트 셔플

날짜	@2021년 8월 6일 → 2021년 8월 7일
링크	
열	
태그	D3 SWEA queue

@2021년 8월 7일 Queue 를 활용하는 방법이 생각났다.



3499 ①

Queue를 활용해보자

n개 들어오는데

$n\%2 == 0$. $idx = \underline{0 \sim N-1}$ $0 \sim 7$ $\underline{0123}$ $\underline{4567}$ $8/2 = 4$

if $\underline{idx < 4}$ Queue 1에 저장

$idx \geq 4$ Queue 2에 저장

$n\%2 == 1$ $idx = 0 \sim N-1$ $0 \sim 8$ $\underline{N=9}$ $\boxed{9/2} = \underline{\underline{4}}$

if $\underline{01234}$ $\underline{5678}$

$idx \leq n/2$ Queue 1에 저장

$idx > n/2$ Queue 2에 저장

출력은 Queue 1 / Queue 2 번갈아 가면서 출력

```
//package d3;
```

```
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.StringTokenizer;

public class Solution {
//public class Test3499_1 {
  public static void main(String[] args) throws NumberFormatException, IOException {
//    System.setIn(new FileInputStream("C:/CodingStudy/SWEA/D3/3499_input.txt"));
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    int T = Integer.parseInt(br.readLine());

    for (int tc = 1; tc <= T; tc++) {
      int num = Integer.parseInt(br.readLine());
      StringTokenizer st = new StringTokenizer(br.readLine());
      String[] line1 = new String[num]; // 1번 Queue
      String[] line2 = new String[num]; // 2번 Queue
      int front1 = 0;
      int front2 = 0;
      int rear1 = 0;
      int rear2 = 0;
      for (int i = 0; i < num; i++) {
        if (num % 2 == 0) { // 홀짝에 따른 Queue 구분 (짝수인 경우)
          if (i < num / 2) // 어디까지 해당 Queue에 저장할지
            line1[front1++] = st.nextToken();
          else
            line2[front2++] = st.nextToken();
        } else { // 홀수인 경우
          if (i <= num / 2)
            line1[front1++] = st.nextToken();
          else
            line2[front2++] = st.nextToken();
        }
      }
      StringBuilder sb = new StringBuilder();
      sb.append("#" + tc + " ");
      for (int i = 0; i < num; i++) {
        sb.append((i % 2 == 0) ? line1[rear1++] + " " : line2[rear2++] + " "); // 번갈아 가면서 출력
      }
      System.out.println(sb);
    }
  }
}
```

@2021년 8월 6일

```
package d3;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.StringTokenizer;
```

```
//public class Solution {
public class Test3499_2 {
  public static void main(String[] args) throws NumberFormatException, IOException {
    System.setIn(new FileInputStream("C:/CodingStudy/SWEA/D3/3499_input.txt"));
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    int T = Integer.parseInt(br.readLine());

    for (int tc = 1; tc <= T; tc++) {
      int num = Integer.parseInt(br.readLine());
      StringTokenizer st = new StringTokenizer(br.readLine());
      String[] line = new String[num];
      for (int i = 0; i < num; i++) {
        line[i] = st.nextToken();
      }
      StringBuilder sb = new StringBuilder();
      sb.append("#" + tc + " ");
      if (num % 2 == 0) {
        for (int i = 0; i < num; i++) {
          sb.append(line[(i % 2 == 0) ? (i / 2) : (i / 2) + num / 2] + " ");
        }
      } else {
        sb.append(line[0] + " ");
        for (int i = 1; i < num; i++) {
          sb.append(line[(i % 2 == 1) ? (i / 2) + num / 2 + 1 : (i / 2)] + " ");
        }
      }
      System.out.println(sb);

    }
  }
}
```