

0919_AS_2주차결과_오윤기

1. 숨바꼭질

```
public class Main1697_숨바꼭질 {
    static int[] di = { -1, 1, 2 };
    static int N, K;
    final static int MAX_SIZE = 100000;

    // 최대값 크기+1 만큼 방문배열을 만들어서 bfs 순회 하면서
    // 동생을 만나게되면 언니가 있는 위치로부터 현재까지온 거리+1
    // 을 return 해준다.
    private static int bfs() {
        Queue<Integer> queue = new LinkedList<Integer>();
        boolean visited[] = new boolean[MAX_SIZE+1];
        queue.add(N);
        visited[N] = true;

        int sec = 0;
        while (!queue.isEmpty()) {
            int size = queue.size();
            for (int s = 0; s < size; s++) {
                int X = queue.poll();
                for (int d = 0; d < 3; d++) {
                    int nx = (d == 2 ? X * di[d] : X + di[d]);
                    if (nx < 0 || nx > MAX_SIZE || visited[nx]) continue;
                    if(nx == K) return sec+1;
                    queue.add(nx);
                    visited[nx] = true;
                }
            }
            sec++;
        }
        return 0;
    }
}
```

2. 전쟁

구현은 못했지만 W와 B를 따로 구분하여 2번 bfs 탐색을 진행하고 1번이라도 탐색된 모든 요소를 카운트하여 결과값을 도출한다.