|  |  |
| --- | --- |
| **BOJ\_10026\_적록색약** | |
| **광주 4반 연주원, 임희선, 전윤철** | |
| **문제 요약** |  |
| * 적록색약이 아닌 사람이 봤을 때 구역의 개수 구하기 * 적록색약인 사람이 봤을 때 구역의 개수 구하기(R, G를 같은 색상으로) | |
| Input | |
|  | 1 <= N <= 100 |
| OutPut | |
| 출력 포맷 확인용  4 3 | 적록색약이 아닌 사람이 봤을 때 구역의 개수와 적록색약인 사람이 봤을 때의 구역의 수를 공백으로 구분 |
| **해결 아이디어**   * 적록색약인 사람의 경우와 아닌 경우를 나누어 그림을 탐색한다.   + 탐색 조건에 따라 2개의 함수를 만든다.   + if문을 이용하여 조건을 구분하여 탐색한다. * R, G, B 값을 2, 1, 0으로 변환하여 적록색약인 사람일 때 양수인 경우와 0인 경우로 나누어 탐색을 진행한다. | |
| **Pseudo-code**  for (int = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  if (방문하지 않은 경우) {  적록색약이 아닌 bfs();  구역의 개수++;  }  }  }  for (int = 0; i < N; i++) {  for (int j = 0; j < N; j++) {  if (방문하지 않은 경우) {  적록색약 bfs();  구역의 개수++;  }  }  }  bfs\_RGB(int r, int c) {  queue 생성  queue에 시작 좌표 넣기  해당 좌표 방문처리  while (!queue.isEmpty()) {  queue에서 좌표 꺼내기  다음 좌표의 범위가 배열 내부에 && 방문하지 않은 좌표 && 이전 좌표의 값과 같을 때  큐에 다음 좌표의 값 넣기  방문 처리  }  }  bfs\_RB(int r, int c) {  if (이전 좌표의 값이 B일 때 다음 좌표의 값이 B인 경우) {  } else {  if (이전 좌표의 값이 B가 아닐 때 다음 좌표의 값이 B가 아닌 경우)  }  } | |