decrease.cpp

[6~7]
배열을 입력받을 벡터 vec, vec의 0,0사이 혹은 시작과 끝 사이의 연속된 값을 받을 벡터 vec2, 길이 n은 입력받을 변수 num, vec.push_back를 위한 변수 temp, 최소 횟수를 나타내는 변수 answer, vec2.size()를 대신하는 변수 coun

[9~14]
길이 n을 매개변수로 받고 for문을 통해 vec에 입력된 원소들을 입력하는 함수 mackVec

[16~38]
num을 매개변수로 입력받고 최소 횟수를 구하는 함수 minimum

길이 n만큼 for문을 도는데,

1 번째 조건으로 vec의 i 번째 원소가 0인지 아닌지 확인한다. 만약 0이 아니라면 coun를 1을 더하고, 해당 원소를 vec2의 push_back 한다.

만약 0이라면 2 번째 조건으로 해당 0 이전에 연속된 자연수의 여부(coun)을 확인한다. 만약 있다면 그 연속된 길이 coun 만큼 vec2에 쌓여있을 테니 min2, max2에 각각 최솟값과 최댓값을 입력한다.

3 번째 조건문으로 만약 최댓값과 최솟값이 같다면 연속된 자연수들은 한 번에 지울 수 있으므로 for 문을 통해 해당 인덱스의 원소들을 전부 0 처리한다. 만약 최솟값과 최댓값이 같지 않다면 해당 인덱스의 원소들을 최솟값만큼 뺀다. 조건문을 거쳤다면 최소 횟수를 나타내는 answer을 1을 더하고 cou, min2, max2, vec2를 다음 연산을 위해 초기화한다.

[40~64]
만약 for 문이 마지막이라면 위와 같은 방식으로 coun의 여부를 확인한 다음 coun이 있다면 아직 처리되지 않은 연속된 자연수들이 있다는 것이므로 최대 최소를 구하고 위에서 처리한 조건문과 같은 방식으로 min2와 max2의 값을 비교한 후 연산한다. 만약 coun이 0이라면 모든 연속된 자연수들이 처리가 된 것이므로 연속된 자연수들을 입력받는 vec2가 아닌 원래의 배열 vec을 조건문을 통해 최댓값이 1 이상이라면 coun을 초기화한 후 꼬리재귀를 하고, 0이라면 배열이 모두 0이 된 것이므로 answer을 리턴한다.